

Humboldt-Universität zu Berlin

Institut für Bibliotheks- und Informationswissenschaft

Bachelorarbeit

# **Konzept und prototypische Erstellung eines Informationssystems auf VuFind-Basis für die Bibliotheks- und Informationswissenschaft**

zur Erlangung des akademischen Grades

Bachelor of Arts (B.A.)

Philosophische Fakultät I

Oliver Pohl (534230)

Dekan: Prof. Michael Seadle, Ph.D.

Gutachter/in: 1. Prof. Vivien Petras, Ph.D.

2. Dipl.-Math. Michael Heinz

Datum der Einreichung: 23.12.2012

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>II</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Listings</b>	<b>V</b>
<b>1 Ease-of-use &amp; Ease-of-access</b>	<b>1</b>
<b>2 Informationssysteme</b>	<b>4</b>
2.1 Definitionen . . . . .	4
2.2 VuFind . . . . .	6
<b>3 LISard</b>	<b>8</b>
3.1 Ziel & Erste Schritte . . . . .	8
3.2 Aufbau der Kollektion . . . . .	9
3.2.1 E-LIS . . . . .	9
3.2.2 UB-HU Berlin . . . . .	11
3.2.3 DNB . . . . .	14
3.2.4 SpringerLink . . . . .	15
3.3 Anpassung des Indexierungsverfahrens . . . . .	18
3.3.1 Mapping . . . . .	18
3.3.2 Stemming . . . . .	19
3.3.3 Stoppwörter . . . . .	19
3.3.4 Boosting & Relevance Ranking . . . . .	20
3.4 Anpassung von Design & Funktionen . . . . .	23
3.4.1 Theme . . . . .	23
3.4.2 Facetten . . . . .	24
3.4.3 Lokale Verfügbarkeit & unnötige Funktionen . . . . .	25
3.4.4 Externe Inhalte, Vorschau & Autovervollständigung . . . . .	27
<b>4 Schlussbetrachtung</b>	<b>29</b>
<b>A Literaturverzeichnis</b>	<b>32</b>
<b>B Andere Quellen</b>	<b>38</b>
<b>C Abbildungen</b>	<b>41</b>
<b>D Listings</b>	<b>52</b>
<b>E DVD</b>	<b>70</b>
<b>Selbstständigkeitserklärung</b>	<b>71</b>

# Abkürzungsverzeichnis

**Abb.** Abbildung

**API** Application Programming Interface

**bzw.** beziehungsweise

**CSS** Cascading Stylesheet

**DDC** Dewey Decimal Classification

**DFG** Deutsche Forschungsgemeinschaft

**d.h.** das heißt

**DisMax** DisjunctionMaxQueryParser

**DNB** Deutsche Nationalbibliothek

**E-LIS** E-prints in Library and Information Science

**engl.** englisch

**FAQ** Frequently Asked Questions

**ggf.** gegebenenfalls

**HTML** Hypertext Markup Language

**HTTP** Hypertext Transfer Protocol

**HU Berlin** Humboldt-Universität zu Berlin

**GBV** Gemeinsamer Bibliotheksverbundml

**GPL** GNU General Public License

**IAO** Fraunhofer Institut für Arbeitswirtschaft und Organisation

**ISBN** International Standard Book Number

**LISard** Library & Information Science - all relevant documents

**MARC** Machine-Readable Cataloging

**OAI-PMH** Open Archive Initiative Protocol for Metadata Harvesting

**OPAC** Online Public Access Catalog

**PDF** Portable Document Format

**PHP** PHP: Hypertext Processor

**PNX** Primo Normalized XML

**Pos.** Position

**RSS** Really Simple Syndication

**RVK** Regensburger Verbundklassifikation

**S.** Seite

**SOLR** Searching On Lucene w/Replication

**SRU** Search/Retrieval via URL

**Tab.** Tabelle

**TF/IDF** Termfrequenz / Inverse Dokumentfrequenz

**ThULB** Thüringer Universitäts- und Landesbibliothek Jena

**u.a.** unter anderem

**UB-HU Berlin** Universitätsbibliothek der Humboldt-Universität zu Berlin

**u.** und

**URL** Uniform Resource Locator

**VSM** Vector Space Model

**XML** Extensible Markup Language

**XSLT** Extensible Stylesheet Language Transformation

**z.B.** zum Beispiel

## Abbildungsverzeichnis

3.1	VuFind im Browser nach der Installation . . . . .	8
3.2	LISard-Logo (Entwurf: Kati Müller) . . . . .	23
C.1	Resultatsliste bei b2i nach einzelnen Datenbanken aufgeschlüsselt . . . . .	41
C.2	Facette bei b2i, um Trefferlisten zusammenzufügen . . . . .	42
C.3	Zusammengefügte Resultatsliste bei b2i . . . . .	43
C.4	Trefferliste auf LISard nach der Indexierung der E-LIS-Datensätze . . . . .	44
C.5	VuFind-Startseite mit dem default-Theme . . . . .	45
C.6	Resultatsliste von VuFind mit dem default-Theme . . . . .	45
C.7	VuFind-Startseite mit angepasstem LISard-Theme . . . . .	46
C.8	Resultatsliste in VuFind mit angepassten LISard-Theme . . . . .	46
C.9	Facettenleiste mit Standardeinstellungen (links), Facettenleiste für LISard angepasst (rechts) . . . . .	47
C.10	Resultatsliste in LISard mit angezeigter Verfügbarkeit und Standort der Treffer . . . . .	48
C.11	Trefferanzeige in LISard mit angezeigter Verfügbarkeit und Standort des Treffers . . . . .	48
C.12	Resultatsliste in LISard ohne angezeigter Verfügbarkeit und Standort der Treffer . . . . .	49
C.13	Trefferanzeige in LISard ohne angezeigter Verfügbarkeit und Standort des Treffers . . . . .	49
C.14	LISard-Startseite ohne die Funktionen: FAQ, Katalog durchstöbern, alpha- betisch Stöbern, Semesterapparat und Frag einen Bibliothekar . . . . .	50
C.15	Trefferanzeige mit Kataloganreicherung in LISard: Geladene Cover (rot), Link zu Google Books (grün) . . . . .	50
C.16	Rechtschreibfehlerkorrektur in LISard . . . . .	51
C.17	Autovervollständigung von Suchanfragen in LISard . . . . .	51

## Listings

1	Angepasste <code>/usr/local/vufind/web/conf/config.ini</code> . . . . .	52
2	Angepasste <code>/usr/local/vufind/harvest/oai.ini</code> für den Harvest von E-LIS . . . . .	52
3	<code>Only-DeEn-oai.sh</code> : Shell-Skript, um nicht deutsch- oder englischsprachige Datensätze aus der E-LIS-Kollektion entfernt . . . . .	52
4	<code>Filter-DeEn-oai.sh</code> : Shell-Skript, das anderssprachige Knoten in den von E-LIS geharvesteten XML-Dateien entfernt . . . . .	53
5	Angepasste <code>e-lis.properties</code> -Datei . . . . .	53
6	Angepasstes <code>e-lis.xsl</code> -Stylesheet . . . . .	53
7	<code>Harvest-Resultlists-AN-Primo.sh</code> : Shell-Skript, das alle Identifier aus einer Trefferliste von Primus filtert (URLs in Variablen zur Übersichtlichkeit gekürzt) . . . . .	54
8	<code>Harvest-PNX.sh</code> : Shell-Skript, das alle Einträge in der Primus-Trefferliste im PNX-Format herunterlädt . . . . .	55
9	<code>Filter-XML-Languages.sh</code> : Shell-Skript, das alle nicht-deutsch- oder englischsprachigen XML-Dateien aus einer Kollektion entfernt und ggf. anderssprachige Knoten aus mehrsprachigen XML-Dateien entfernt . . . . .	55
10	<code>primus.xsl</code> : XSLT-Stylesheet für PNX-Datensätze . . . . .	56
11	<code>Harvest-DNB.sh</code> : Shell-Skript, das MARC21-XML-Datensätze vom Katalog der DNB herunterlädt . . . . .	59
12	<code>marc.properties</code> : MARC-Feld 008/35-37 soll nicht indexiert werden, DNB als Institution eintragen (Ausschnitt), Building entfernen . . . . .	59
13	<code>Harvest-SpringerLink-Resultlists.en.sh</code> : Shell-Skript, das alle Links zu Einträgen aus einer Resultatsliste von SpringerLink herunterlädt . . . . .	59
14	<code>Harvest-SpringerLink-Articles.sh</code> : Shell-Skript, das HTML-Quelltexte von SpringerLink analysiert, Metadaten extrahiert und diese in einer XML-Datei speichert . . . . .	60
15	<code>springerlink.xsl</code> : XSLT-Stylesheet für die Zeitschriftenartikel von SpringerLink . . . . .	63
16	<code>format_map.properties</code> : Angepasste Translation Map für Formate . . . . .	65
17	<code>schema.xml</code> : Auswahl des Stemmers . . . . .	66
18	<code>facets.ini</code> : Gegenüberstellung der voreingestellten und für LISard angepassten <code>facets.ini</code> . . . . .	66
19	Auskommentierte Stellen in den Dateien <code>result.tpl</code> und <code>holdings.tpl</code> . . . . .	67
20	<code>footer.tpl</code> : Bestimmte Recherche- und Hilfsfunktionen deaktivieren . . . . .	68
21	<code>config.ini</code> : Buch-Cover und weitere Informationen laden . . . . .	69
22	<code>searches.ini</code> : Autovervollständigung für Suchanfragen einschalten . . . . .	69

## 1 Ease-of-use & Ease-of-access

In ihrer Metastudie “Why are Online Catalogs Hard to Use“ fasste Borgman 1986 die Probleme von Benutzern verschiedener bibliographischer Retrieval Systeme und Online Kataloge zusammen: Viele, insbesondere auch unerfahrene Nutzer hatten Schwierigkeiten mit Retrieval Systemen umzugehen. Die Queries einiger Nutzer führten aufgrund von Rechtschreibfehlern bei der Formulierung der Suchanfrage zu keinem Ergebnis. Ein viel grundlegenderes Problem stellte jedoch die mangelnde Kenntnis im Umgang mit den Retrievalsystemen (“Conceptual Knowledge“) dar. So waren z.B. boole’sche Verknüpfungen oder Trunkierungen, die Suchanfragen präzisieren und dadurch auch genauere Suchergebnisse liefern können, unbekannt.

Hollnagel u. Woods (1983) schlagen vor, dass Designer von „Man-Machine-Systems“ darauf abzielen sollen, ihre Systeme auf die Fähigkeiten der Nutzer abzustimmen. Dazu merkt Marchionini (1995, S. 75) an, dass es sich dabei um ein Henne-Ei-Problem handelt:

„We cannot discover how users can best work with systems until the systems are built, yet we should build systems based on knowledge of users and how they work.“

Durch die voranschreitende Entwicklung des Internets und seiner Möglichkeiten konnte man über die letzten Jahrzehnte einen Wandel in der Benutzung von Retrieval-Systemen beobachten. Vor der Jahrtausendwende lehnten sich Internet-Suchmaschinen an die Prinzipien damaliger Information-Retrieval-Systeme an (Jansen et al., 1998). Heutzutage erwarten Nutzer „traditioneller“ Information-Retrieval-Systeme (z.B. OPACs) Gegebenheiten, die sie von aktuellen Suchmaschinen kennen (Sadeh, 2007a); sie erwarten „ease-of-use, ease-of-access and speed that characterize internet tools and services.“

Immer mehr Bibliotheken in Deutschland setzen Systeme ein, die den Ansprüchen der heutigen Nutzer gerecht werden sollen. So ist das „Discovery & Delivery Solution Primo“ von ExLibris bereits in 10 deutschen wissenschaftlichen Bibliotheken, unter anderem auch an der Humboldt-Universität zu Berlin<sup>1</sup>, im Einsatz.<sup>2</sup>

Auch im Fachinformationsbereich werden Systeme wie Primo eingesetzt: Aus dem Hause der HU Berlin entstammt die virtuelle Fachbibliothek EVIFA<sup>3</sup>, welche auch auf Primo basiert und dessen Vorteile bietet. Ein zentraler Vorteil von Primo ist der integrierte Index, welcher Datensätze aus verschiedenen Datenquellen in sich vereint, homogenisiert und somit alle vorhandenen Datensätze mit einer Suchanfrage durchsuchbar macht.

---

<sup>1</sup><http://primus.ub.hu-berlin.de>

<sup>2</sup>Stand August 2011. ExLibris selbst hat bei einer Anfrage per eMail keine Auskunft gegeben, wie viele Bibliotheken bereits selbst nutzen. Die Information stammt aus einem persönlich geführten Interview mit Herrn Michael Voß, welcher u.a. für die technische Implementierung von Primo an der Universitätsbibliothek der Humboldt-Universität zu Berlin (UB-HU Berlin) mitverantwortlich war.

<sup>3</sup>[www.evifa.de](http://www.evifa.de)

Die Resultate aus heterogenen Datenressourcen werden dem Nutzer als eine gerankte, homogene Liste präsentiert, ohne dass er „Detailwissen über [...] den Ablageort bestimmter Daten besitzen muss“. (Dietz et al., 2007)

Die virtuelle Fachbibliothek für Bibliotheks- und Informationswissenschaft b2i<sup>4</sup> verfolgt einen anderen Ansatz. Bei einer Suchanfrage werden zwar Daten verschiedener Quellen durchsucht, allerdings werden die Ergebnisse der Suche standardmäßig als separate Listen, jeweilig nach Datenquelle geordnet, angezeigt. Während also bei EVIFA die Suche mit Primo dem Prinzip einer Websuche („googlen“) nachgeht (Sadeh, 2007a), bricht b2i mit jenem Prinzip. Wenn auch beide Portale den Designrichtlinien für „Search User Interfaces“ nach Hearst (2009) größtenteils entsprechen, fehlen b2i essenzielle Funktionen wie z.B. das Hervorheben von Suchtermen in der Resultatsliste, Rechtschreibkorrektur oder Autovervollständigung von Suchanfragen.

In dieser Arbeit wird die Erstellung und Konfiguration eines prototypischen Fachinformationsportals für die Bibliotheks- und Informationswissenschaft auf VuFind-Basis dokumentiert. Es wurde ein Rechercheeinstieg mit einem integrierten Index erstellt, der Datensätze verschiedener fachrelevanter Quellen in sich vereint, und so die Anzeige von Ergebnissen bei Suchanfragen in einer vereinten Resultatsliste ermöglicht. Ziel war es, potentiellen Nutzern ein System anbieten zu können, dessen Benutzerführung sie schon von anderen Suchmaschinen und Retrievalsystemen kennen und gewohnt sind (Novotny, 2004), und das die von Sadeh (2007a) erwähnte Erwartungshaltung der Nutzer an solche Systeme (ease-of-use, ease-of-access and speed) erfüllt.

Der zu dokumentierende Prototyp eines Fachinformationssystems kann jedoch nicht den Ansprüchen virtueller Fachbibliotheken gerecht werden und „alle relevanten Informationen und Dokumente zu einem Fachgebiet“, abbilden (Meyenburg, 2000). Die Implementierung von Funktionen zur unmittelbaren Beschaffung von Medieneinheiten oder Funktionen zur interaktiven Kommunikation mit Nutzern (z.B. Chatbots) werden außen vor gelassen und stellen potentiell ein eigenes Thema für eine Haus- oder Bachelorarbeit dar.

In dieser Arbeit soll demonstriert werden, dass der Aufbau eines Informationssystems bzw. Fachinformationsportal mit einem integrierten Index und einer einheitlichen Trefferliste auf VuFind-Basis machbar ist. Der Schwerpunkt liegt dabei auf der Anpassung der Recherchefunktionen des Fachinformationssystems.

Im zweiten Kapitel dieser Arbeit wird auf den Begriff „Informationssystem“ eingegangen und im Gebiet der Fachinformation eingeordnet. Zusätzlich wird die Open Source OPAC-Software VuFind sowie dessen Anwendungen in Deutschland vorgestellt.

Kapitel 3 dokumentiert die Erstellung des prototypischen Fachinformationsportals für die Bibliotheks- und Informationswissenschaft *LISard* (*Library & Information Science*

---

<sup>4</sup>[www.b2i.de](http://www.b2i.de)



- *all relevant documents*). Zunächst wird eine Testkollektion für LISard erstellt und indexiert. Anschließend werden die Änderungen am Indexierungsverfahren und Design nähergebracht. Weiterhin wird die Implementierung von Funktionserweiterungen bzw. Addons beschrieben.

In Kapitel 4 wird eine kritische Schlussbetrachtung vorgenommen. Es ist zu untersuchen, welche Funktionen noch zusätzlich implementiert werden sollten und wo Verbesserungspotential in LISard zu finden ist. Weiterhin soll ein Vergleich zwischen b2i und LISard gezogen werden. Am Ende soll geprüft werden, ob das Ziel, den Komfort eines integrierten Index und einer einheitlichen Trefferliste, eingehalten werden konnte und wo sich LISard aus den in Kapitel 2.1 vorgestellten Definitionen einordnen lässt.

## 2 Informationssysteme

### 2.1 Definitionen

Noch bevor die Deutsche Forschungsgemeinschaft (DFG) 1998 in ihrem Memorandum über die Weiterentwicklung der überregionalen Literaturversorgung die Gründung von virtuellen Fachbibliotheken empfahl, halfen Web-Kataloge, Linklisten oder Subject Gateways Nutzern beim Recherchieren nach wissenschaftlichen Quellen im Internet (Rösch u. Weisbrod, 2004). Rösch (2001) klassifiziert diese Dienste inklusive Web-Kataloge als „internetspezifische Suchwerkzeuge [der ersten Generation]“; Meta-Suchmaschinen zählt er zur zweiten Generation. Portale, die Rösch der dritten Generation internetspezifischer Suchwerkzeuge zuordnet, werden von Kirchhof et al. (2006, S. 5) folgendermaßen definiert:

„Ein Portal ist definiert als eine Applikation, welche basierend auf Webtechnologien einen zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse darstellt. Charakterisierend für Portale ist die Verknüpfung und der Datenaustausch zwischen heterogenen Anwendungen über eine Portalplattform. Eine manuelle Anmeldung an den in das Portal integrierten Anwendungen ist durch Single-Sign-On nicht mehr notwendig, es gibt einen zentralen Zugriff über eine homogene Benutzungsoberfläche.“

Schellhase (2008, S. 175) scheint mit seiner Definition für Fachinformationsportale an Kirchhof et al. inhaltlich anzuknüpfen. Nach ihm „[bieten] Fachinformationsportale Zugänge zu fachspezifischen Publikationen, [...] können unterschiedliche bibliographische Systeme integrieren und somit dazu beitragen, die bestehende Zersplitterung von fachlichen Nachweissystemen zu überwinden.“

Virtuelle Fachbibliotheken können als Fachinformationsportale verstanden werden: sie sind ebenso „medienübergreifend strukturiert und ermöglichen Zugriff auf für das jeweilige Forschungsfeld relevante Informationen“ (Wissenschaftsrat, 2012, S. 37). Im Gegensatz zu generischen Fachinformationsportalen, sind virtuelle Fachbibliotheken als „digitale Dienste für die seit 1950 [von der DFG] geförderten Sondersammelgebiete“ zu betrachten.

Sowohl virtuelle Fachbibliotheken als auch Fachinformationsportale sind (*Fach*)-*Informationssysteme*; beide erfüllen die vorgeschlagenen Kriterien von Taylor u. Joudrey (2009, S. 159), die ein Informationssystem ausmachen: 1. *storage* (Organisation von Daten), 2. *retrieval* (Suchen und Durchsuchen von Daten anhand von Anfragen) sowie 3. *display* (Bereitstellung eines Interfaces und Präsentation der Daten). Die Begriffe „System, Informationssystem und Information Retrieval System“ können im informationswissenschaftlichen Kontext synonym gebraucht werden.

Betrachtet man die Aufgabenfelder von Fachinformationsportalen, so liegt deren Hauptaugenmerk auf der „Integrationsfunktionalität, d.h. der Aggregation von wissenschaftlichen

Informationen und deren unmittelbarer Beschaffung, sowie auf der Recherche“ (Guba, 2006).

Die virtuelle Fachbibliothek für den Bereich der Bibliotheks- und Informationswissenschaft ermöglicht zwar die von Guba beschriebene Informationsaggregation für dieses Fach, allerdings scheitert das Informationssystem an der von Schellhase geschilderten Überwindung der Zersplitterung fachlicher Nachweissysteme: Die Ergebnisliste zu einer Suchanfrage wird bei b2i nach den einzelnen durchsuchten Datenbanken aufgeschlüsselt - im Grunde sind die Informationssysteme weiterhin zersplittert (siehe Abb. C.1, S. 41). Am Ende der Facettenleiste am linken Bildschirmrand existiert eine Möglichkeit (siehe Abb. C.2, S. 42), die verschiedenen Trefferlisten zu einer einzigen zusammenfügen zu lassen. Erst dann (Abb. C.3, S. 43) ergibt sich eine vereinheitliche Trefferliste.

Für den Nutzer ist es wünschenswert, von vornherein eine Trefferliste zu erhalten, in der Resultate aus allen Quellen enthalten und nach Relevanz geordnet sind. Laut Höchstötter u. Lewandowski (2009) sehen sich die Hälfte aller Nutzer von Suchmaschinen nur die erste Seite der Resultatsliste an und formulieren anschließend ihre Anfrage neu, sollten sie keinen für sie relevanten Treffer gefunden haben. Ein b2i-Nutzer müsste verhältnismäßig weit auf der Webseite herunterscrollen, um eine vereinte Trefferliste anzeigen zu lassen. Da die meisten Nutzer nach zehn bis 15 Treffern aufhören zu scrollen, ist es wahrscheinlich, dass einem Großteil der Nutzer die Option zur vereinten Trefferliste gar nicht erst auffällt.

Wäre die Resultsliste von Anfang an vereint, müsste der Nutzer nicht erst die geeignete Datenbank ausfindig machen (Dietz et al., 2007) und kann auf das zurückgreifen, was er von anderen Suchmaschinen, z.B. Google oder Yahoo!, gewohnt ist (Sadeh, 2007a).

Innerhalb dieser Arbeit wurde ein prototypisches Informationssystem auf VuFind-Basis erstellt, das den Komfort einer einheitlichen, integrierten Trefferliste bieten soll.

## 2.2 VuFind

VuFind (gesprochen „View Find“) ist eine 2007 an der Falvey Memorial Library der Villanova University (Pennsylvania, USA) entwickelte plattformunabhängige Open Source Bibliotheksportalsoftware (Kinstler, 2010). Durch die Anwendung von VuFind wird den Betreibern von Bibliotheks- und Informationsportalbetreibern ermöglicht, einen integrierten Index zu schaffen. Da VuFind unter der GPL-Lizenz steht, ist es jeder Person gestattet, VuFind herunterzuladen, zu benutzen, anzupassen und weiter zu verteilen (Houser, 2009).

Die Suchfunktion VuFinds baut auf Apache SOLR und damit auch auf der Lucene Search Library auf.<sup>5</sup> Somit ist es mit VuFind möglich, verschiedenste Dateiformate im Volltext zu indexieren und durchsuchbar zu machen. Betreiber können einzelne Parameter in den Algorithmen nach ihrem Belieben konfigurieren und so ein angepasstes Informationssystem nach ihren Vorstellungen kreieren.

In Deutschland erfreut sich VuFind wachsender Beliebtheit. Derzeit wurde die Software u.a. bereits an der TUB Harburg<sup>6</sup>, Staatsbibliothek Hamburg<sup>7</sup> und der Universitätsbibliothek Leipzig<sup>8</sup> implementiert.<sup>9</sup> Die Universitätsbibliothek Kassel plant, VuFind 2013 einzuführen (Dröbler, 2012).

Neben institutioneller Implementierungen von VuFind fällt das studentische Projekt „bachelopac“<sup>10</sup> von Schmitt u. Stehle (2010) auf. Im Rahmen einer Bachelorarbeit erstellen sie ein VuFind-basiertes Bibliothekssystem, das auf dem Katalog des Hochschul- und Bibliotheksservice der HAW Hamburg aufbaute. Ein weiteres studentisches Projekt stellt „GenderVu“<sup>11</sup> von Zierold (2011) dar, welches mittels VuFind eine neue Rechercheoberfläche für die Genderbibliothek<sup>12</sup> bereitstellt.

Das Projekt „digitalesthueringen.de“<sup>13</sup> von der Thüringer Universitäts- und Landesbibliothek Jena (ThULB) kann als Beispiel betrachtet werden, auch Themen- oder Fachportale mit VuFind zu gestalten. Dieses Portal „bietet einen zentralen Sucheinstieg zu einer Vielzahl digitalisierter Kulturgüter des Freistaats Thüringen.“<sup>14</sup>

Die ThULB und GenderVu haben gezeigt, dass eine „Zweckentfremdung“ VuFinds, abseits von der Nutzung als Bibliothekskatalog, möglich ist. Diese Bachelorarbeit knüpft an jene Projekte an und dokumentiert die Erstellung eines prototypischen Fachinformati-

---

<sup>5</sup><http://lucene.apache.org/solr/>

<sup>6</sup><https://katalog.tub.tu-harburg.de/>

<sup>7</sup><http://beluga.sub.uni-hamburg.de/vufind/>

<sup>8</sup><http://www.ub.uni-leipzig.de/>

<sup>9</sup>Eine größere Auflistung deutscher VuFind-benutzender Institutionen ist hier zu finden: [http://vufind.org/wiki/installation\\_status](http://vufind.org/wiki/installation_status)

<sup>10</sup><http://46.163.77.244/vufind/>

<sup>11</sup><http://www2.gender.hu-berlin.de/genderbib/gendervu/>

<sup>12</sup><http://www.gender.hu-berlin.de/genderbib>

<sup>13</sup><http://www.digitalesthueringen.de/tportal>

<sup>14</sup><http://www.digitalesthueringen.de/tportal/HTML/Home?page=About>

onsportals für die Bibliotheks- und Informationswissenschaft. VuFind erscheint für die Erstellung eines solchen Portals aufgrund seiner eigens definierten Ziele geeignet:

„The goal of VuFind is to enable your users to search and browse through all of your library’s resources by replacing the traditional OPAC to include:

- Catalog Records
- Digital Library Items
- Institutional Repository
- Institutional Bibliography
- Other Library Collections and Ressources“<sup>15</sup>

VuFind kann unterschiedliche Datenquellen integrieren und mit heterogenen Dateiformaten, ob Metadaten oder Volltexten bzw. andere digitale Medientypen, umgehen. Für die Erstellung eines Fachinformationsportals ist es wichtig, möglichst relevante Daten verschiedener relevanter Ressourcen zu implementieren.

---

<sup>15</sup><http://vufind.org/about.php>

## 3 LISard

### 3.1 Ziel & Erste Schritte

Unter dem Arbeitstitel „Library & Information Science - all relevant documents (LISard)“ wurde die Erstellung eines Fachinformationsportals mit einem integrierten Index angestrebt, welcher alle Daten durchsucht und die relevanten Ergebnisse in einer Trefferliste wiedergibt. Die LISard-Instanz wird auf einem PC mit Ubuntu Quantal Quetzal (12.10)<sup>16</sup> mit VuFind 1.3 betrieben. Das VuFind-Team stellt auf ihrer Website eine ausführliche Dokumentation für die Installation und Anpassung von VuFind bereit.<sup>17</sup>

Die Erstinstallation von VuFind gestaltet sich insbesondere für Ubuntu-Nutzer einfach. Auf der Download-Seite von VuFind<sup>18</sup> befindet sich direkt ein `.deb`-Paket zur direkten Installation zur Verfügung. Innerhalb weniger Mausklicks bzw. Terminalbefehle kann VuFind auf der lokalen Maschine installiert werden. Nach der Installation befindet sich die VuFind-Instanz im Ordner: `/usr/local/vufind`.

Damit das lokale VuFind vom eigenen Webbrowser angesteuert werden kann, muss zunächst ein Parameter in der Datei `/web/conf/config.ini` geändert werden (siehe Listing 1, S. 52). Anschließend kann VuFind mittels `./vufind.sh start` gestartet und im Browser aufgerufen werden. Abb. 3.1 zeigt einen Screenshot einer neu installierten VuFind-Maschine.

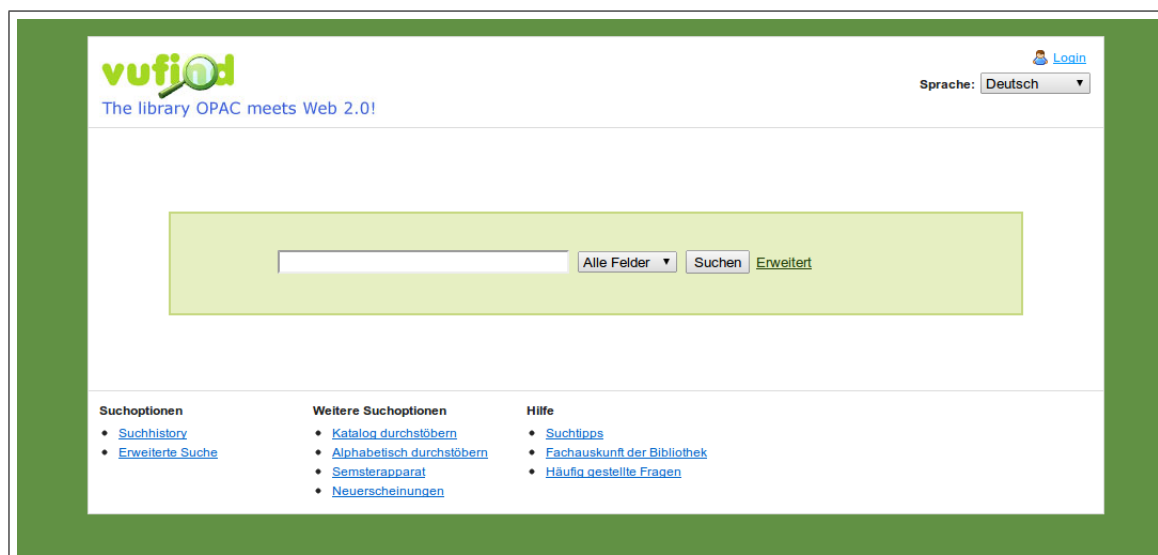


Abbildung 3.1: VuFind im Browser nach der Installation

<sup>16</sup>[http://wiki.ubuntuusers.de/Quantal\\_Quetzal](http://wiki.ubuntuusers.de/Quantal_Quetzal)

<sup>17</sup>[http://vufind.org/wiki/vufind\\_1.x\\_user\\_manual](http://vufind.org/wiki/vufind_1.x_user_manual)

<sup>18</sup><http://vufind.org/downloads.php>

## 3.2 Aufbau der Kollektion

Um den angestrebten Portalcharakter dieser VuFind-Instanz gerecht zu werden, wurde eine Kollektion aus verschiedenen Ressourcen bzw. Ressourcentypen (siehe Tab. 1) gewählt, deren bibliotheks- und informationswissenschaftliche Datenbestände es zu harvesten und in LISard zu indexieren galt.

Ressourcentyp	Ressource	Website
Open Access Repository	E-LIS	<a href="http://eprints.rclis.org/">http://eprints.rclis.org/</a>
Bibliothekskatalog	UB-HU Berlin	<a href="http://primus.ub.hu-berlin.de">http://primus.ub.hu-berlin.de</a>
Bibliographie	DNB	<a href="http://www.dnb.de">www.dnb.de</a>
Verlagsangebot	SpringerLink	<a href="http://link.springer.com/">http://link.springer.com/</a>

Tabelle 1: Ressourcen für LISard

Um das breite Spektrum von in Frage kommenden Datenbanken prototypisch abzubilden, wurden Metadaten von einem Open Access Repository (E-LIS), einem Bibliothekskatalog (UB-HU Berlin), einer bibliographischen Datenbank (DNB) sowie einem Verlagsangebot (SpringerLink) gesammelt. Der folgende Abschnitt geht genauer auf den Harvesting-Prozess der einzelnen Angebote und deren Integration in LISard ein. Der Datenbestand wird auf deutsch- und englischsprachige Publikationen eingeschränkt.

Eine Indexierung ist von folgenden Dateien bzw. Dateitypen abhängig:

- `schema.xml` - Gibt an, welche Metadatenfelder im Index existieren
- `[Quelle].properties` - Spezifiziert, wie die zu Indexierenden Metadaten indexiert werden sollen
- ggf. `[Quelle].xsl` - XSLT-Stylesheet zur Transformation von zu indexierenden XML-Datensätzen (im Zusammenspiel mit `[Quelle].properties`)
- Kollektion an zu indexierenden Datensätzen

### 3.2.1 E-LIS

E-prints in Library and Information Science (E-LIS) ist ein Open Access Repository für den Fachbereich der Bibliotheks- und Informationswissenschaft. Wissenschaftler können ihre fachrelevanten Publikationen auf E-LIS unter einer Open-Access-Lizenz hinterlegen und ihre Veröffentlichungen für andere verfügbar machen (Morrison et al., 2007).

Wie die meisten Open Access Repositories verfügt E-LIS über eine OAI-PMH-Schnittstelle, welche das Harvesting von Metadaten für andere Dienste ermöglicht. VuFind

stellt Skripte und Funktionen für das Sammeln von Metadaten über OAI-PMH im Verzeichnis folgenden Verzeichnis bereit: `/usr/local/vufind/harvest/`. Die PHP-Datei `harvest_oai.php` greift auf die Datei `oai.ini` zurück, welche Parameter für den Harvestingvorgang, wie z.B. die URL der OAI-PMH-API vorgibt. Listing 2 (S. 52) zeigt die für den Harvest von E-LIS angepasste `oai.ini`-Datei. Führt man das PHP-Skript aus, werden alle Datensätze von E-LIS im Dublin Core-Format heruntergeladen. Insgesamt wurden 14277 Datensätze über die OAI-PMH-Schnittstelle heruntergeladen. Da auch bereits auf E-LIS entfernte bzw. nicht mehr zugreifbare Datensätze importiert werden, kann die Datenmenge mit dem Shell-Skript `batch-delete.sh` bereinigt werden.

Weil lediglich deutsch- und englischsprachige Datensätze in den späteren Index geladen werden sollen, müssen anderssprachige Datensätze herausgefiltert werden. Mittels des Kommandozeilentools `xmlstarlet`<sup>19</sup>, welches zur Bearbeitung und Transformation von XML-Dateien genutzt werden kann, werden unerwünschte Datensätze aus der zu indexierenden Kollektion entfernt (siehe Listing 3, S. 52). Ein zweites Shell-Skript (siehe Listing 4, S. 53) bearbeitet verschiedensprachige Datensätze. Manche Publikationen auf E-LIS wurden in verschiedenen Sprachen veröffentlicht, allerdings sollen im entstehenden LISard-Index keine anderen Sprachen außer Deutsch und Englisch verzeichnet sein. Dazu werden XML-Nodes aus den betroffenen Dateien entfernt.

Anschließend können die bereinigten Daten indexiert werden. Um zu bestimmen, welche Metadaten und wie diese indexiert werden sollen, müssen zuvor noch ein XSLT-Stylesheet und eine `.properties`-Datei<sup>20</sup> für die Daten von E-LIS erstellt werden. VuFind liefert bereits entsprechende Dateien für DSpace-basierte Repositorien wie E-LIS. Mitarbeiter der Naval Postgraduate School (2012) zeigen, wie die Konfigurationsdateien anzupassen sind, um Datensätze von DSpace-Repositorien erfolgreich mit VuFind zu indexieren. Für den Datenimport von E-LIS muss lediglich eine Zeile in der entsprechenden XSLT-Datei geändert werden (siehe Listing 6, S. 53). Die `properties`-Datei gibt die Parameter für die Indexierung, wie z.B. die Auswahl des XSLT-Stylesheets oder den Namen der zu indexierenden Kollektion, an. Das XSLT-Stylesheet legt fest, wie die Daten aus den XML-Datensätze ausgelesen werden sollen. Ließe man das Stylesheet unverändert, könnten die Identifier der einzelnen Datensätze nicht korrekt von VuFind geparkt und aufgerufen werden.

Nun können die bereinigten Daten mit folgendem Befehl importiert werden:

```
bash vufind/harvest/batch-import-xsl.sh E-LIS e-lis.properties
```

<sup>19</sup>`xmlstarlet` kann, falls nicht auf dem System vorhanden, mit `sudo apt-get install xmlstarlet` nachinstalliert werden.

<sup>20</sup>Die Dateien befinden sich im Verzeichnis: `/usr/local/vufind/import/e-lis.properties` (siehe Listing 5, S. 53) bzw. `/usr/local/vufind/import/xsl/e-lis.xsl`



Die VuFind-Dokumentation zum Thema Datenimport empfiehlt, nach der abgeschlossenen Indexierung die VuFind-Instanz neu zu starten und den Index mittels eines bereitgestellten PHP-Skripts<sup>21</sup> zu optimieren. Abbildung C.4 (S. 44) zeigt einen Ausschnitt der Trefferliste von LISard, in der alle bisher indexierten Daten ausgegeben werden.

*Zusammenfassung:*

1. Datensätze harvesten (`oai.ini` und `harvest_oai.php`)
2. Bereits auf E-LIS gelöschte Daten lokal entfernen (`batch-delete.sh`)
3. Deutsch- und englischsprachige Datensätze filtern (`Only-DeEn-oai.sh` und `Filter-DeEn-oai.sh`)
4. `e-lis.properties` und `e-lis.xsl` erstellen und anpassen
5. Indexierung starten: `bash vufind/batch-import-xsl.sh E-LIS e-lis.properties`

### 3.2.2 UB-HU Berlin

Die Universitätsbibliothek der Humboldt-Universität zu Berlin führte 2012 die „Discovery & Delivery Solution Primo“ (ExLibris, 2012) unter den Namen „Primus“ als Recherchesystem für den Regelbetrieb im Bibliotheksalltag ein und ersetzte damit den bisherigen OPAC als primäres Suchwerkzeug.

Der größte Unterschied zu „herkömmlichen“ OPACs ist, dass Primo auf einen integrierten Index zurückgreift. Bibliotheken speisen ihre Datenbestände sowie, wenn gewünscht, die Datenressourcen dritter Anbieter in Primos zentralen Index (Sadeh, 2007b). Vor dem Indexieren werden die aus heterogenen Quellen stammenden und in verschiedenen Formaten vorliegenden Datensätze normalisiert, und im PNX-Format (Primo Normalized XML) im Index hinterlegt (Hänger et al., 2010). Da nun alle Datensätze in einem einheitlichen Format vorhanden sind, kann in allen integrierten Ressourcen über eine Suchmaske recherchiert werden (ExLibris, 2009).

Die Gegebenheit, dass alle Datensätze im PNX-Format vorliegen, macht Primo-Kataloge für einen Harvest attraktiv. Wird im Browser ein Eintrag auf Primo aufgerufen, muss lediglich die Zeichenkette `&showPnx=true` an die URL angehängen werden, um den PNX-Datensatz des aufgerufenen Eintrags zu erhalten. Dieser Umstand kann genutzt werden, um automatisiert PNX-Datensätze aus einem Primo-Katalog herunterzuladen.

<sup>21</sup>`/usr/local/vufind/util/optimize.php`

Für den LISard-Korpus wird exemplarisch die Universitätsbibliothek der Humboldt-Universität zu Berlin gewählt, da sie für das Institut für Bibliotheks- und Informationswissenschaft der HU Berlin ein großer Bestand bibliotheks- und informationswissenschaftlicher Literatur führt.

Zunächst gilt es, alle relevanten, d.h. bibliotheks- und informationswissenschaftlichen Einträge im Katalog der UB-HU Berlin ausfindig zu machen. Da Primus keine Möglichkeit bietet, direkt nach Notationen im Katalog zu suchen, wird ein Umweg eingeschlagen. In Primus wird nach „Information“ gesucht und ein relevanter Treffer ausgewählt. Über die RVK-Notation in den bibliographischen Angaben des ausgewählten Eintrags gelangt man zu weiteren Treffern mit der selben Notation. Analysiert man die URL zu allen Treffern einer RVK-Notation, fällt der String `„v1(freeText0)=AN+10000&v1(173155009UI0)=lsr19&v1(421939160UI1)=all_items&v1(1UI0)=exact&fn=search“` auf. Dieser Teil des HTTP-Requests, um alle Datensätze einer RVK-Notation aufzurufen, könnte sich folgendermaßen interpretieren lassen<sup>22</sup>:

- `v1(freeText0)=AN+10000` - Der Suchterm ist „AN 10000“
- `v1(173155009UI0)=lsr19` - Suche im Bereich RVK-Notation
- `v1(421939160UI1)=all_items` - Suche im ganzen Datenbestand
- `v1(1UI0)=exact` - Es soll nach dem exakten Suchterm gesucht werden
- `fn=search` - Funktion: Suche ausführen

Für LISard kommen alle Datensätze mit dem RVK-Notationspräfix „AN“ für „Buch- und Bibliothekswesen, Informationswissenschaft“ in Frage. Dementsprechend wird der Suchterm mit einer Wildcard (\*) versehen, um alle gewünschten Datensätze ausgeben zu lassen: `v1(freeText0)=AN+*`.

Der nächste Schritt ist, zu jedem Datensatz zu navigieren und die jeweilige PNX-Entsprechung herunterzuladen. Für diesen Zweck wurden zwei Shell-Skripte verfasst. Das erste Skript (siehe Listing 7, 54) analysiert die HTML-Quelltexte jeder einzelnen Seite der Trefferliste nach Identifikatoren zu den Einträgen und speichert die gefunden Identifier in einer Liste. Das zweite Skript (siehe Listing 8, S. 55 ) greift auf diese Liste zurück, navigiert zu jedem Suchtreffer und lädt die dazugehörige PNX-Datei herunter.

Nachdem der Download abgeschlossen ist, sollen, wie im Kapitel 3.2.1, nicht-deutsch- oder englischsprachige Datensätze aus der Kollektion bzw. Einträge in den PNX-Dateien,

---

<sup>22</sup>Diese Annahme ist das Ergebnis verschiedener Abwandlungen des originalen HTTP-Requests. Einzelne Parameter wurden aus der Anfrage gelöscht, um eine abgewandelte Trefferliste zu erhalten und um damit auf die Funktionen der einzelnen Parameter schließen zu können.

die auf eine Anderssprachigkeit hinweisen, entfernt werden. Zu diesem Zweck wurde ein Shell-Skript programmiert, das eben jene Aufgabe erfüllt (siehe Listing 9, S. 55).

Die übrig gebliebenen Datensätze sollen nun in den LISard-Index eingespeist werden. Für die Indexierung von XML-Dateien benötigt VuFind ein auf die XML-Datensätze angepasstes XSLT-Stylesheet (siehe Listing 10, S. 56) und die dazugehörige **properties**-Datei. Tabelle 2 schlägt vor, wie sich PNX-Dateien auf die von VuFind bereitgestellten XSLT-Stylesheetvorlagen mappen lassen könnten.

XSLT-Feld	XPath-Ausdruck bei PNX
ID	//recordid
Language	//facets/language
Subject	//search/subject
Type	//display/type
Author	//display/contributor
Title	//display/title
Publisher	//display/publisher
Publishdate	//display/creationdate
URL	//sourcerecordid

Tabelle 2: Mapping von PNX zu einem XSLT-Stylesheet (Ausschnitt)

Sind ein passendes XSLT-Stylesheet (Listing X) und eine angepasste **properties**-Datei erstellt, kann die Primus-Kollektion indexiert werden.

```
bash vufind/harvest/batch-import-xsl.sh Primus primus.properties
```

*Zusammenfassung:*

1. Datensätze herunterladen mit der RVK-Notation „AN \*“ (Harvest-Resultlists-AN-Primo.sh und Harvest-PNX.sh)
2. Anderssprachige Dateien entfernen und PNX-Datensätze aufbereiten (Filter-XML-Languages.sh)
3. PNX nach XSLT mappen; **primus.xsl** und **primus.properties**
4. Indexierung starten: `bash vufind/batch-import-xsl.sh Primus primus.properties`

### 3.2.3 DNB

Das Gesetz über die Deutsche Nationalbibliothek (DNBG, 2006) beauftragt die DNB, alle „ab 1913 in Deutschland veröffentlichten [und] ab 1913 im Ausland veröffentlichten deutschsprachigen Medienwerke, Übersetzungen deutschsprachiger Medienwerke in andere Sprachen und fremdsprachige Medienwerke über Deutschland im Original zu sammeln, zu erschließen und bibliographisch zu verzeichnen ...“ Das Pflichtexemplarrecht gibt die Abgabe zweier Exemplare jeder entsprechenden Veröffentlichung an die DNB vor. Es spielt keine Rolle, ob es sich bei der Publikation um Drucksachen, Mikroformen, audiovisuelle Medien oder körperlose, d.h. digitale, Medien handelt (Seefeldt u. Syré, 2007).

Der Katalog bzw. OPAC der DNB<sup>23</sup> kann auch als Bibliographie verstanden werden, da die bibliographische Verzeichnung, also die Führung eines Existenznachweises von Medienwerken, eine der Hauptaufgaben der Deutschen Nationalbibliothek ist.

Zu fast jedem Katalogeintrag bietet die DNB die zugehörigen Metadatenätze im MARC21-XML-Format kostenfrei zum separaten Download an. Wollen Bibliotheken, Archive oder andere Einrichtungen Metadatenpakete von der DNB beziehen, können sie diese in verschiedenen Metadatenformaten über den Datenshop der DNB<sup>24</sup> käuflich erwerben. Ein Testzugang zum Datenshop der Deutsche Nationalbibliothek (2012) ermöglicht einen kostenlosen Download von 100 Datensätzen; möchte man mehr herunterladen, wird eine Gebühr verlangt. Da knapp 10.000 Datensätze von der DNB für einen Download in Frage kommen, beliefe sich der Preis auf über 1.000 Euro.

Da der Kauf von Metadaten den preislichen Rahmen für das LISard-Projekt übersteigt, wurde ein Shell-Skript verfasst, welches zu jedem vorhandenen Eintrag mit der DDC-Notation „02\*“ für Bibliotheks- und Informationswissenschaft navigiert und, sofern verfügbar, den dazugehörigen Metadatenatz im MARC21-XML-Format herunterlädt (siehe Listing 11, S. 59).

Es kann nicht ausgeschlossen werden, dass sich unter den heruntergeladenen Daten auch nicht deutsch- oder nicht englischsprachige Datensätze befinden. Um den Import unerwünschter Metadaten in LISard zu vermeiden, müssen anderssprachige Datensätze bzw. Einträge herausgefiltert werden. Primäre Voraussetzung für eine Aufnahme in den LISard-Index für einen Datensatz ist in deutscher oder englischer Sprache vorzuliegen. Die MARC-Felder 008/35–37 und 041\$a in den heruntergeladenen Datensätzen geben an, welche Sprachen im jeweiligen Medium vorkommen (Library of Congress, 2010 und 2012).

Zur Filterung wird auf das Shell-Skript (Listing 9, S. 55) aus Kapitel 3.2.2 zurückgegriffen. Um die MARC21-XML-Dateien für das Shell-Skript nutzbar zu machen, müssen die erste und letzte Zeile jeder Datei entfernt werden, da sonst das XML-Toolkit `xmlstarlet`

---

<sup>23</sup>[www.dnb.de](http://www.dnb.de)

<sup>24</sup><https://portal.dnb.de/metadataShop.htm>

nicht auf die Knoten in der Datei zugreifen kann. Das Shell-Skript entfernt nicht deutsch- oder nicht englischsprachige Datensätze aus dem Verzeichnis und bereinigt im Feld 041\$a mehrsprachige Einträge, sodass am Ende nur noch „ger“ und „eng“ als Werte in jenem Feld vorhanden sind (siehe Listing 11, S. 59). Einige Datensätze weisen zusätzlich das MARC-Feld 041\$h auf, welches die Originalsprache eines Medienwerkes angibt.

Damit keine unerwarteten Überschneidungen zwischen den MARC-Feldern 008/35-37 sowie 041\$h und 041\$a auftreten, werden erstere vom Indexierungsverfahren ausgeschlossen. Das Solrmarc-Wiki (2010) beschreibt den Aufbau und die Syntax einer generischen `index.properties`-Datei, die die Parameter für die Indexierung von MARC-Dateien bestimmt. In diesem Fall wird die `marc.properties`-Datei im Verzeichnis `/usr/local/vufind/import` abgeändert (siehe Listing 12, S. 59).

In der gleichen Datei wird der Parameter `institution = "MyInstitution"` zu `institution = "Deutsche Nationalbibliothek"` verändert, sodass später bei LISard die hier zu indexierenden Datensätze als von der DNB (als Instanz) stammend gekennzeichnet werden. Des Weiteren wird der Parameter `building`, welcher den lokalen Standort der zum Datensatz zugehörigen Medieneinheit spezifiziert, aus der `marc.properties`-Datei entfernt, weil LISard als Fachinformationsportal keine physischen Bestände führt.

Sind die Datensätze alle aufbereitet worden, können sie mit folgendem Befehl indexiert werden:

```
bash vufind/harvest/batch-import-marc.sh DNB
```

*Zusammenfassung:*

1. Datensätze im Bereich `ddc=02*` von der DNB herunterladen (`Harvest-DNB.sh`)
2. Anderssprachige Dateien entfernen und MARC21-XML-Dateien aufbereiten (`Filter-DNB.sh`)
3. `marc.properties` anpassen (MARC-Feld 008/35-37 und 041\$h vom Indexieren ausschließen, Parameter `institution` umbenennen und Parameter `building` entfernen)
4. Indexierung starten: `bash vufind/batch-marc-import.sh DNB`

### 3.2.4 SpringerLink

SpringerLink (2012b) ist das Onlineangebot des Julius-Springer-Verlags und bezeichnet sich selbst als „einen der führenden Online-Informationendienste für naturwissenschaftliche,

technische und medizinische Bücher und Zeitschriften“. Im Angebot von SpringerLink<sup>25</sup> ist zwar keine explizit bibliothekswissenschaftliche, dafür aber informationswissenschaftliche Literatur gelistet.

Den Katalog an eJournals und eBooks und in denen enthaltenen Beiträgen bzw. Kapiteln schlüsselt SpringerLink nach Disziplinen auf. Für LISard kommen Einträge aus Disziplinen wie „Database Management & Information Retrieval“, „Information Systems & Applications“ sowie „Human Computer Interaction“ in Frage.

Metadatenätze zu eBooks können kostenfrei im MARC-Format von über ein haus-eigenes Download-Tool<sup>26</sup> bezogen werden (SpringerLink, 2012a). Über das Download-Tool wurden insgesamt 9502 Datensätze heruntergeladen.

Das Tool bietet jedoch keine Downloadmöglichkeit für Datensätze von Zeitschriftenartikeln. Um auch Springer-Zeitschriften bzw. -Zeitschriftenartikel in den LISard speisen zu können, wurde ähnlich wie in Kapitel 3.2.2 vorgegangen. Erneut wurden zwei Shell-Skripte verfasst. Das erste Skript (siehe Listing 13, S. 59) analysiert die Liste aller Einträge in der Disziplin „Database Management & Information Retrieval“ und filtert die jeweiligen Links heraus.<sup>27</sup>

Das zweite Skript (Listing 14, S. 60) greift die Links auf, extrahiert Metadaten aus den HTML-Quelltexten und speichert die gewonnen Metadaten in einer XML-Datei.

Die MARC-Datei, welche Metadaten für eBooks von SpringerLink beinhaltet, kann bereits indexiert werden:

```
bash vufind/import-marc.sh [MARC-Datei].mrc
```

Für die heruntergeladenen XML-Dateien der SpringerLink-Zeitschriftenartikel wird ein angepasstes XSLT-Stylesheet benötigt. Mit dem im Listing 15 (S. 63) gezeigten Stylesheet lassen sich die XML-Dateien erfolgreich in den LISard-Index übernehmen;

```
bash vufind/harvest/batch-import-xsl.sh SpringerLink springerlink.properties
```

#### *Zusammenfassung:*

- MARC-Kollektion von SpringerLink herunterladen

<sup>25</sup><http://link.springer.com/>

<sup>26</sup><http://www.springer.com/?referer=springer.com&SGWID=1-148802-3020-0-0>

<sup>27</sup>Um die Komplexität dieser Arbeit einzuschränken, wurde der Harvest auf eine Disziplin beschränkt. Die Liste aller Einträge ist hier zu finden: <http://link.springer.com/search?just-selected-from-overlay=facet-sub-discipline&query=&just-selected-from-overlay-value=%22Database+Management+%26+Information+Retrieval%22&facet-discipline=%22Computer+Science%22&facet-sub-discipline=%22Database+Management+%26+Information+Retrieval%22>

- MARC-Datei indexieren: `bash vufind/import-marc.sh [MARC-Datei].mrc`
- Metadaten zu Zeitschriftenartikeln im Bereich „Data Management & Information Retrieval“ extrahieren (`Harvest-SpringerLink-Resultlists.sh` und `Harvest-SpringerLink-Articles.sh`)
- Passendes XSLT-Stylesheet zum erhaltenen Ausgabeformat (`springerlink.xsl`) und dazugehörige `springerlink.properties`-Datei verfassen
- Indexierung starten: `bash vufind/harvest SpringerLink springerlink.properties`
- Index optimieren: `php vufind/util/optimize.php`

Nach Abschluss der Indexierung der Datensätze aus allen vier geharvesteten Ressourcen befinden sich insgesamt 38949 Einträge im Index von LISard (siehe Tabelle 3). Um den Index zu optimieren, wird VuFind neu gestartet und ein entsprechendes PHP-Skript (`optimize.php`) gestartet.

Ressource	Anz. Datensätze
SpringerLink	16939
UB-HU Berlin	9676
DNB	7490
E-LIS	4844
Insgesamt	38949

Tabelle 3: Anzahl indexierter Datensätze nach Ressource

### 3.3 Anpassung des Indexierungsverfahrens

Wird ein Dokument mit VuFind bzw. SOLR indexiert, wird das Dokument zuerst tokenisiert, d.h. der Text als Bitstrom wird in einzelne Sätze und Sätze in einzelne Wörter unterteilt (Schmid, 2009). Diese Tokens können dann von SOLRs Indexierungsmechanismen weiterverarbeitet werden (SOLR-Wiki, 2012a). Die einzelnen Tokens werden auf ihren Wortstamm reduziert (gestemmt) und mit Stoppwortlisten abgeglichen, um semantisch unbedeutsame Terme herauszufiltern. Einzelne Terme oder Felder können geboostet werden - ihnen wird eine größere Bedeutung zugemessen als anderen Termini bzw. Felder. Bei einer Suchanfrage mit einem der geboosteten Terme werden Dokumente, die eben jene Termini enthalten, höher gerankt. Des Weiteren können bei der Indexierung Werte einzelner Felder untereinander gemappt, d.h. zugeordnet, werden. Dies ist unter anderem nötig, um Datensätze unterschiedlicher Herkunft und Formate zu vereinheitlichen.

#### 3.3.1 Mapping

Aufgrund der Heterogenität der Metadaten in LISard existieren für manche Felder unterschiedliche Bezeichnungen (Repräsentationen eines Begriffs), obwohl derselbe Begriff gemeint ist. Im Feld, in dem die Sprache des Datensatzes angegeben wird, können Werte wie „De, de, deu, ger“ vorkommen; im Formatfeld können für Bücher die Einträge „book, Book, buch, Buch“ auftreten.

Damit alle Bezeichnungen dem richtigen Begriff zugeordnet werden, können in VuFind Translation-Maps konfiguriert werden. In den Translation-Maps wird festgelegt, welche Werte innerhalb eines Datenfelds untereinander gruppiert werden können. Für jede Gruppe an unterschiedlichen Werten wird ein String definiert, der alle Werte in einer Gruppe global repräsentiert. Dieser Prozess wird auch als „Mapping“ bezeichnet. Listing 16 (S. 65) zeigt einen Ausschnitt der Translation-Map, die für die richtige Zuordnung von Formaten zuständig ist.

Für LISard wurden vor allem die Translation-Maps für Sprache und Formate editiert. Zusätzlich wäre es lohnenswert, alle Themen oder Schlagwörter zu mappen. Durch die Gruppierung und Zuordnung von Schlagwörtern und Themen werden die Ähnlichkeiten zwischen den einzelnen Datensätzen erhöht. Erweiterte man LISard um weitere Datenbanken, könnte per Mapping das Problem der Inter-Indexer-Konsistenz („Maß an Übereinstimmungen zwischen unterschiedlichen Indexaten der selben Vorlage verschiedener Indexer“, Stock, 2008, S. 358) entschärft werden: Das gleiche Dokument, von unterschiedlichen Personen/Systemen indexiert, erhielte bei der Indexierung in LISard einheitliche Schlagwörter bzw. Themenzuordnungen. Aufgrund der hohen Anzahl an Schlagwörter und Themen, wurden diese jedoch nicht gemappt.



### 3.3.2 Stemming

Information Retrieval Systeme nutzen Stemmer, um variierende Wortformen auf ihren Stamm zu reduzieren und somit einen besseren Abgleich zwischen einer Suchanfrage und den indexierten Dokumenten vornehmen zu können (Xu u. Croft, 1998). Die Zusammenführung verschiedener Wortformen (conflate) zu einem Stamm (stemming) bezeichnet Krovetz (1993) als gängiges Verfahren im Information Retrieval.

Der erste Stemming-Algorithmus wurde von Lovins (1968) geschaffen und verfügt über 35 „recording rules“ sowie 294 Suffixe mit je 29 kontextabhängigen Regeln zur Wortstammreduzierung in der englischen Sprache. Porter (1980) kreierte einen simpleren Algorithmus mit ca. 60 Suffixen, zwei „recording rules“ und einer kontextabhängigen Regel. Mittlerweile wird der Algorithmus von Porter als Standard für das Stemming der englischen Sprache (Willett, 2006) genutzt.

Über die Jahre entwickelte Porter (2006) Stemming-Algorithmen für weitere europäische Sprachen, darunter auch Deutsch. Die Snowball-Algorithmen von Porter sind häufig als voreingestellte Stemmer in SOLR-betriebenen Suchmaschinen festgelegt.

Von Hause aus nutzt VuFind den englischen Snowball-Stemmer von Porter. Um einen anderen Stemmer auszuwählen muss die Datei `vufind/solr/biblio/conf/schema.xml` abgeändert werden. Neben der Vorgabe, welche Felder im Index existieren sollen, gibt die `schema.xml`-Datei Parameter für die Indexierung an.

Porter hat für die deutsche Sprache zwei Stemming-Algorithmen geschrieben. Der wesentliche Unterschied zwischen den beiden ist, dass der zweite im Vergleich zum ersten Algorithmus Umlaute als Doppelvokale (ä → ae) interpretiert (SOLR-Wiki, 2012b).

Für LISard wird der Snowball-Stemmer „German2“<sup>28</sup> ausgewählt, sodass mehr relevante Treffer bei Suchanfragen, die Umlaute enthalten, ausgegeben werden können. Listing 17 (S. 66) zeigt die nötige Änderung für die Auswahl des Stemmers.

### 3.3.3 Stoppwörter

Während der Indexierung werden die Terme in einem Dokument bzw. Datensätze nicht nur gestemmt, sondern auch mit einer Stoppwortliste abgeglichen. Stoppwortlisten schließen Wörter von der Indexierung aus, die schlechte Indexterme wären (Fox, 1989). Wird ein Dokument indexiert kann jeder darin enthaltene Term als potentieller Indexterm betrachtet werden (Wilbur u. Sirotkin, 1992). Dabei kommen einige dieser Terme häufig in diesem und in anderen Dokumenten vor, und haben höchstens eine grammatikalische, jedoch keine semantische Bedeutung für das zu indexierende Dokument (Salton, 1989, S. 279).

<sup>28</sup><http://snowball.tartarus.org/algorithms/german2/stemmer.html>

Savoy (2005c) stellt Stoppwortlisten für verschiedene Sprachen bereit, die nach dem von Fox vorgestellten Schema erstellt wurden: Zuerst werden alle Wörter in einem Korpus nach Häufigkeit sortiert gelistet und die ersten 200 davon genauer analysiert. Zahlen sowie semantisch bedeutungsvoll erscheinende Wörter werden dabei aus der Liste entfernt. Den übrig gebliebenen Wörtern in der Liste werden Possessivpronomen, Präpositionen, Konjunktionen sowie Abkürzungen in der jeweiligen Sprache hinzugefügt, sodass am Schluss eine für die jeweilige Sprache gültige Stoppwortliste generiert wird.

VuFind bzw. SOLR hat im Auslieferungszustand zwar eine Stoppwortliste implementiert, allerdings ist diese sehr kurz und nur für die englische Sprache ausgelegt. Von daher wurde auf die von Savoy zur Verfügung gestellten deutsch- und englischsprachigen Stoppwortlisten zurückgegriffen und zu der von SOLR hinzugefügt (Savoy, 2005a & 2005b).

Die ursprünglich 33 Wörter enthaltende Stoppwortliste im Verzeichnis `/vufind/solr/biblio/conf/stopwords.txt` wurde auf insgesamt 1115 Wörter in Deutsch und Englisch erweitert.

### 3.3.4 Boosting & Relevance Ranking

Lucene, damit auch SOLR und VuFind, nutzen als Retrievalmethode eine Kombination aus dem Vector Space Model (VSM) (Salton et al., 1975) und dem Boole'schem Retrieval (Lucene-Wiki, 2012). VuFind nutzt zusätzlich den „DisjunctionMaxQueryParser“ (DisMax) (VuFind-Dokumentation, 2012d), welcher eine Vereinfachung der LuceneQueryParser-Syntax darstellt. Im Vergleich zum LuceneQueryParser, bietet der DisMax weniger Funktionalitäten. Ein Großteil von Sonderzeichenoperationen wie Trunkierungen (\*,!,?) oder Boole'schen Operatoren (AND, OR, NOT) werden bei DisMax nicht als solche interpretiert (Hostetter, 2010).

Demnach eignet sich der normale Query Parser von Lucene besser für professionelle Recherchen, während der DisMaxQueryParser eher für einfache Recherchen ohne Operatoren ausgelegt ist.

Wird eine Suche in VuFind ausgeführt, rankt SOLR die Treffer unter anderem nach TF/IDF-Kriterien (Salton et al., 1983; Lucene-Wiki, 2012):

- „Je seltener ein Suchwort im Index vorhanden ist, desto höher ist sein Beitrag zum Ranking.
- Je länger der Feldinhalt ist, in dem ein Suchwort gefunden wurde, desto geringer ist sein Beitrag zum Ranking.
- Dokumente, in denen Suchworte häufiger vorkommen, werden höher bewertet.

- Dokumente, in denen mehr Suchworte vorkommen, werden höher bewertet.“
- Einzelne Dokumente, Felder, Suchworte oder Phrasen können mit einem Boostfaktor versehen werden. Geboostete Dokumente bzw. Dokumente, die geboostete Felder, Suchworte oder Phrasen enthalten, werden höher gerankt. (Langenstein u. Maylein, 2009; Schmitt u. Stehle, 2010)

Ein Ziel von LISard als Fachportal ist es, seine Nutzer auf möglichst relevante Treffer mit möglichst direktem Zugriff auf Volltexte zu verweisen. Deshalb sollen Datensätze, die einen Link zu einer elektronischen Ressource mit ggf. Volltextzugriff beinhalten, in LISard höher gerankt werden. Da der größte Teil der bibliotheks- und informationswissenschaftlichen Ressourcen in Zeitschriften und Konferenzen geschieht, sollen dementsprechende Einträge auch höher gerankt werden.

Die Boostfaktoren sollen folgendermaßen vergeben werden:

- Dokumente, die Zeitschriftenartikel oder Konferenzbeiträge sind und zu denen jeder Nutzer einen Volltextzugriff hat, erhalten einen Boost von 2.0.
- Dokumente, die Zeitschriftenartikel oder Konferenzbeiträge sind, jedoch keinen Volltextzugriff für jeden Nutzer möglich ist, erhalten einen Boost von 1.5.
- Dokumente, zu denen jeder Nutzer Volltextzugriff hat, jedoch keine Zeitschriftenartikel oder Konferenzbeiträge sind, erhalten einen Boost von 1.3.

Um Boostfaktoren bei der Indexierung von Dokumenten festzulegen, werden für XML-basierte Datensätze die entsprechenden XSLT-Stylesheets angepasst. Der Tag `<doc>` wird ein Attribut für den Document Boost hinzugefügt: `<doc boost="1.5">`. Bei der vorhandenen Kollektion gestaltet sich die Verteilung der Boost-Faktoren relativ simpel.

Die von SpringerLink heruntergeladenen Daten im XML-Format sind alles Metadaten-sätze für Zeitschriftenartikel und erhalten somit einen Document Boost von 1.5. Alle von E-LIS geharvesteten Datensätze verweisen auf das Open Access Repository. Die dahinterliegenden Dokumente sind für jeden Nutzer frei zugänglich, sodass alle E-LIS-Datensätze automatisch mindestens einen Document Boost von 1.3 oder maximal 2.0 erhalten. Da es mit XSLT nicht möglich ist, über `xsl:if` oder `xsl:choose` nur einen Opening Tag zu erzeugen, wird die E-LIS Kollektion geteilt. Der erste Teil enthält nur Zeitschriftenartikel und Konferenzbeiträge; der zweite Teil den Rest. Für jede Teilkollektion wird dann das XSLT-Stylesheet und die `e-lis.properties`-Datei angepasst, um den passenden Boostfaktor bei der Indexierung übertragen zu können.

Auf Datensätze von Primo wird kein Boost angewandt, da weder Dokumente im Volltext auf Primus hinterlegt sind, noch Zeitschriftenartikel in dieser Kollektion enthalten sind. Ein

Document Boost von MARC- bzw. MARC21-XML-Dateien scheint bei der Indexierung nicht möglich zu sein. Selbst der Chefentwickler VuFinds Katz (2011) beschreibt den Import von MARC-Datensätzen als „hässlich“.

Sind die XSLT-Stylesheets angepasst, können die vorhandenen Daten reindexiert werden. Zuvor muss der bereits vorhandene Index im Verzeichnis `/vufind/solr/biblio/index` gelöscht werden. Anschließend können alle Datensätze wie im Unterkapitel 3.2 erneut eingespeist werden. Bei dieser Indexierung werden die im aktuellen Unterkapitel vorgenommenen Änderungen am Indexierungsverfahren (Mapping, Stemming, Stoppwörter, Boosting) berücksichtigt.

### 3.4 Anpassung von Design & Funktionen

Nachdem alle Dokumente mit dem angepassten Indexierungsverfahren aufgenommen wurden, ist die Arbeit mit den Datensätzen an sich abgeschlossen. Der Fokus liegt nun auf der Anpassung der Benutzeroberfläche, darunter die Auswahl eines neuen Layouts, Platzierung und Auswahl von Facetten sowie die Entfernung unnützer Angaben in der Trefferliste.

#### 3.4.1 Theme

Im Auslieferungszustand bietet VuFind vier verschiedene Themes (graphische Oberflächen) zur Implementierung an. Als Standard ist das Theme „blueprint“ installiert, welches auch auf Abb. 3.1 (S. 8) und C.4 (S. 44) zu sehen ist.

Die Auswahl des anzuwendenden Themes kann in der `config.ini`-Datei (siehe Listing 1, 52) ausgewählt werden, welche bereits in Kapitel 3.1 besprochen wurde. LISard soll ein blaues Theme erhalten, da diese Farbe Seriösität (Seeborn, 2011) und Professionalität (Wikibooks, 2011) repräsentiert. Nach dem Vergleich aller Themes wurde entschieden, ein angepasstes LISard-Theme auf Grundlage des „default“-Themes (siehe Abb. C.5, und C.6, S. 45) zu erstellen.

Zum Namen LISard wurde ein passendes Logo (siehe Abb. 3.2) entworfen. Dieses soll eine Echse (engl. Lizard) enthalten und in einem Blauton gehalten sein. Das Blau vom Logo soll auch maßgeblich für das restliche Design sein. Dafür wurde eine Kopie des default-Themes angelegt und zu LISard umbenannt. In diesem Ordner wurden dann die ursprünglichen VuFind-Logos durch das neue LISard-Logo ersetzt. Des Weiteren wurden CSS-Dateien editiert, um die Farbgebung von Grün auf Blau zu bewirken.



Abbildung 3.2: LISard-Logo (Entwurf: Kati Müller)

Abschließend wird in der `config.ini` das LISard-Theme ausgewählt, sodass fortan die getroffenen Designentscheidungen beim Aufruf von LISard angezeigt werden (siehe Abb. C.7 und C.8, S. 46).

*Zusammenfassung:*

1. Theme auswählen (`config.ini`)
2. Design, Farben & Logos anpassen (`/vufind/web/interface/themes/LISard`)

### 3.4.2 Facetten

Bei einer Suche in VuFind werden automatisch Facetten für alle Treffer gebildet, gruppiert und angezeigt. Der Nutzer sieht, wie viele Treffer in welchen Formaten, aus welcher Datenquelle oder welcher Zeitspanne stammen. In einer Eye-Tracking-Studie von Kules et al. (2009) wurde beobachtet, wie lange Nutzer auf bestimmte Bereiche in einem OPAC schauen. Im Durchschnitt verweilten die Nutzer ca. 25 Sekunden auf den Facetten und 50 Sekunden auf der Resultatsliste.

Ramdeen u. Hemminger (2011) untersuchten, ob Studenten „herkömmliche“ Recherchesysteme (ohne Facetten) oder Recherchesysteme mit Facetten bevorzugen. Sie fanden heraus, dass die Mehrheit der Studienteilnehmer facettierte Interfaces präferierten. Zu den meist angeklickten Facetten gehörten „subject, format and author“.

Harter (1986, S. 244) definiert Facetten als eine Konzeptgruppe („concept group“), die aus gleichen oder verwandten Termen in einer Kategorie besteht. Wird eine Facette ausgewählt (z.B. Sprache: Englisch) werden nur noch Treffer in englischer Sprache in der Resultatsliste erscheinen. Mit diesem Klick wird eine boole'sche Verknüpfung (Query AND Language=eng) ausgelöst (English et al., 2002).

Die angezeigten Facetten werden bei VuFind abhängig von den gefundenen Treffern generiert. Liegen also Informationen zu Autoren oder Themen vor, werden die entsprechenden Facetten angezeigt; sind keine vorhanden, erscheinen sie nicht. Für das Fachinformationsportal ist es nicht zwingend notwendig, die Signatur oder Klassifikation einer Medieneinheit anzeigen zu lassen, zumal diese aufgrund der Heterogenität der Datensätze nicht einheitlich und nicht gemappt bzw. nicht vorhanden sind.

Um die Reihenfolge in der Seitenleiste angezeigter Facetten zu ändern kann die Datei `facets.ini` im Verzeichnis `/vufind/web/conf/` editiert werden (VuFind-Dokumentation, 2012f). Die Reihenfolge der in dieser Datei gelisteten Facettenparameter beeinflusst die Ordnung der Facettenanzeige im Browser. Die verfügbaren Facetten sind abhängig von der `schema.xml` von SOLR, die vorgibt, welche Felder für eine Indexierung zur Verfügung stehen.

Tabelle 4 (S. 25), Listing 18 (S. 66) und Abbildung C.9 (S. 47) zeigen eine Gegenüberstellung der voreingestellten und der für LISard angedachten Facettensortierung. Die Anzahl möglicher Facetten wurde von ursprünglich zehn auf sechs reduziert. Für Nutzer ergibt sich der Vorteil, dass ein grober Überblick über die Struktur der Treffer (Antelman et al., 2006) mit weniger Scrolling (Höchstötter u. Lewandowski, 2009) gewonnen werden kann.

Die Facetten für Genre (z.B. Belletristik, Sachbuch etc.) sowie Ära (z.B. Klassik, Renaissance etc.) und Region (z.B. Deutschland, Europa, Asien etc.) sind für die Bibliotheks- und Informationswissenschaft aufgrund ihrer relativ jungen Geschichte und Ortsunge-

Pos.	Standard	LISard
1	Institution	Ressource (Institution umbenannt)
2	Gebäude	Format
3	Format	Autor
4	Signatur	Sprache
5	Autor	Schlagwörter
6	Sprache	Zeitspanne eingrenzen
7	Genre	
8	Ära	
9	Region	
10	Zeitspanne eingrenzen	

Tabelle 4: Gegenüberstellung der voreingestellten und für LISard angepasste Facettenanzeige

bundenheit verhältnismäßig irrelevant. Viel wichtiger ist die Anzeige von Schlagworten (subjects). Damit diese Facette angezeigt wird, wurde die Datei `facets.ini` um den Parameter `topic = Subject` erweitert.

Auf einem handelsüblichen HD-ready-Monitor mit einer Auflösung von 1280x720 Pixeln passen nach Änderderung der `facets.ini` alle Facetten auf einen Bildschirm, ohne dass gescrollt werden muss.

*Zusammenfassung:*

1. Reihenfolge der Facetten ändern (`facets.ini`)
2. Auswahl der Facetten festlegen: Ressource, Format, Autor, Sprache, Schlagwörter, Zeitspanne (`facets.ini`)

### 3.4.3 Lokale Verfügbarkeit & unnötige Funktionen

Werden MARC-Datensätze in einen Index überspielt, geht VuFind davon aus, dass eine dem Datensatz zugehörige Medieneinheit in den lokalen Bestand der VuFind betreibenden Institutionen übernommen wurde. LISard verfügt jedoch über keinen eigenen lokalen Bestand, sondern verweist lediglich auf die Ressourcen, von denen die Treffer stammen.

Werden Treffer aus der DNB-Kollektion oder aus dem SpringerLink-e-Book-Kontingent gefunden, werden diese von LISard als lokal verfügbar angezeigt. Als Standort wird eine generische „3rd Floor Main Library“ sowie eine generische Signatur „A1234.567“ angezeigt (siehe Abb. C.10, und C.11, S. 48).

Um diese unnötige Information in der Trefferliste und Trefferansicht auszublenden, müssen `.tpl`-Dateien aus dem Verzeichnis `/vufind/web/interface/themes/LISard/RecordDrivers/Index/` bearbeitet werden. Diese Dateien sind dafür verantwortlich, dynamisch Werte aus einem indexierten Datensatz aufzurufen und in einen HTML-Quelltext zu integrieren, sodass ein Browser eine Liste gefundener Treffer bzw. eine Trefferansicht selbst mit den passenden Metadaten anzeigen kann.

Die betroffenen Stellen in den Dateien `result.tpl` (Resultatsliste) und `holdings.tpl` (Trefferanzeige) wurden auskommentiert (siehe Listing 19, S. 67), sodass jene Zeilen nicht geparkt und keine Standortangabe oder Signatur in der Resultatsliste bzw. Trefferanzeige dargestellt werden (siehe Abb. C.12 und C.13, S. 49).

Einen weiteren Sucheinstieg, den VuFind bietet, sind „Stöbern“-Funktionen. Wählt man diese aus, kann man ohne vorher eine Suchanfrage abgeschickt zu haben den Index anhand der zur Verfügung stehenden Facetten einschränken und so eine Trefferliste generieren. Diese Funktion ist allerdings nicht mit den für LISard indexierten Daten funktional. Diese und Funktionen wie das Durchsuchen von Semesterapparaten oder eine Kontaktaufnahme zu einem Bibliothekar erscheinen für LISard im derzeitigen Stadium von geringem Nutzen und wurden daher deaktiviert.

Weitere Rechercheeinstiege wie alphabetisches Stöbern (nach z.B. Themen oder Verfassern) oder Neuerscheinungslisten sind für LISard sinnvoll. Um eine gute Benutzung für die alphabetische Stöber-Funktion zu gewährleisten wäre es erforderlich, vorhandene Schlagwörter zu mappen. Um eine Neuerscheinungsliste zu pflegen, wäre ein regelmäßiger Harvest von Datenquellen und eine möglichst automatisierte Indexierung von Nöten. Da die Pflege jener Funktionen den Rahmen dieser Arbeit übersteigt, werden sie (temporär) deaktiviert. Das FAQ (Häufig gestellte Fragen) beinhaltet im Auslieferungszustands VuFinds nichts und wird daher auch ausgeblendet. Lediglich die Funktion „Suchtipps“ bleibt erhalten, da diese allgemeingültige Hinweise zur Recherche mit VuFind gibt.

Die genannten Recherchefunktionen werden in der Datei `footer.tpl` deaktiviert (siehe Listing 20, S. 68). Abbildung C.7 (unten, S. 46) zeigt die Startseite LISards, auf welcher die Funktionen noch gelistet sind; Abbildung C.14 (S. 50) zeigt die gleiche Seite ohne diese Funktionen.

*Zusammenfassung:*

1. Verfügbarkeit, Standort und Signatur in der Resultatsliste deaktivieren (`result.tpl`)
2. Verfügbarkeit, Standort und Signatur in der Trefferanzeige deaktivieren (`holdings.tpl`)



3. Funktionen: Katalog durchstöbern, alphabetisch stöbern, Semesterapparat, Neuerscheinungen, Häufig gestellte Fragen und frag einen Bibliothekar deaktivieren (`footer.tpl`)

#### 3.4.4 Externe Inhalte, Vorschau & Autovervollständigung

Während des Harvestingprozesses in Kapitel 3.2 wurden nur bibliographische Metadaten mit ggf. Links zu Volltexteten oder gescannten Inhaltsverzeichnissen heruntergeladen. Sonstige Daten, die zur Kataloganreicherung LISards beitragen könnten, fehlen.

VuFind stellt Funktionen bereit, um in dieser Situation Abhilfe schaffen zu können. Über Schnittstellen zu u.a. Google Books, HathiTrust oder OpenLibrary können zusätzliche Daten heruntergeladen werden, um die Datensätze im eigenen Katalog zu erweitern. Da bei der Anreicherung derzeit nur ISBNs übermittelt werden können, funktioniert dieser Prozess nur für Bücher bzw. e-Books. (VuFind-Dokumentation, 2012b und 2012e).

Über die genannten Dienste können u.a. Cover, Rezensionen und Inhaltsangaben bzw. -verzeichnisse von Büchern heruntergeladen werden. Obwohl nur wenige Nutzer von Next-Generation OPACs die Einblendung von Covern als nützlich empfinden, macht es das Informationssystem für die Nutzer attraktiver in der Benutzung (Tam et al., 2009).

Liegt auf Google Books ein einsehbares Digitalisat von einem Buch, wird automatisch ein Button in VuFind generiert, der zum Buch auf Google Books weiterleitet. Abbildung C.15 (S. 50) zeigt einen Ausschnitt einer Trefferliste in LISard, wo der Unterschied zwischen Treffern mit und ohne Datenanreicherungen deutlich wird.

Um diese Features zu aktivieren, muss erneut die `config.ini`-Datei editiert werden. Die betroffenen Zeilen können in Listing 21 (S. 69) eingesehen werden.

Neben diesen nutzerfreundlichen Funktionen bietet VuFind andere Funktionalitäten, die im alltäglichen Gebrauch von Suchmaschinen selbstverständlich geworden sind. Die Nutzung von Rechtschreibkorrektur (siehe Abb. C.16, S. 51) und das Hervorheben von Suchtermen in der Resultatsliste sind bei VuFind voreingestellt. Zusätzlich kann eine Autovervollständigung aktiviert werden (VuFind-Dokumentation, 2012a), die bei der Eingabe einer Suchanfrage dem Nutzer mögliche Queries für seine Recherche oder sogar Treffer in der Suchmaske vorschlägt (siehe Abb. C.17, S. 51).

Die Autovervollständigung lässt sich aktivieren, indem man die im Listing 22 (S. 69) angegebenen Parameter der Datei `searches.ini` im Verzeichnis `vufind/web/conf/` editiert.

*Zusammenfassung:*

1. Kataloganreicherung aktivieren (`config.ini`)
2. Buch-Cover-Anzeige aktivieren (`config.ini`)
3. Autovervollständigung in der Suchmaske aktivieren (`searches.ini`)

## 4 Schlussbetrachtung

Zu Beginn dieser Arbeit wurde das Ziel ausgegeben, die Machbarkeit der Erstellung eines Fachinformationssystem mit einem integrierten Index und einer vereinten Resultatsliste auf VuFind-Basis zu zeigen.

Es wurde eine VuFind-Instanz installiert und auf die Anforderungen an ein Fachinformationsportal für die Bibliotheks- und Informationswissenschaft zugeschnitten. Dabei wurden Datensätze in heterogenen Datenformaten aus vier unterschiedlichen Quellen gesammelt, gefiltert und in einen integrierten Index importiert. Weiterhin erfolgten Anpassungen am Indexierungsverfahren und an der graphischen Oberfläche. Während der Durchführung dieser Arbeit lag das Hauptaugenmerk auf den Retrieval- und Recherche-funktionen VuFinds.

Als Resultat dieser Arbeit liegt ein funktionsfähiges Fachinformationsportal namens LISard für den bibliotheks- und informationswissenschaftlichen Bereich vor, welches mehrere Ressourcen in einem Index integriert und durchsuchbar macht. Bei einer Suchanfrage wird eine Trefferliste generiert, die Treffer aus allen verschiedenen Ressourcen beinhaltet.

Vor Beginn dieser Arbeit wurde eine Evaluation von LISard angedacht. Aus Kapazitätsgründen wurde die Evaluation wieder verworfen. Die Evaluation und Optimierung von Fachinformationsportalen stellt ein eigenes Thema für eine Bachelorarbeit dar. Es wäre interessant gewesen, potentielle Nutzer einem Xtreme Discount Usability Test (Marty u. Twidale, 2005) zu unterziehen, um die größten Probleme im Umgang mit LISard zu eruieren. Wären diese Probleme behoben, könnte man zusätzlich asynchrone Nutzertests durchführen, bei dem Nutzer orts- und zeitunabhängig (Greifeneder, 2012, S. 13) einer Aufgabe mit LISard nachgehen. Aus den Logfiles und ggf. Nutzerkommentaren ließe sich eine Einsicht gewinnen, wie mit LISard umgegangen wird.

Ein weiterer wünschenswerter Aspekt, der nicht implementiert werden konnte, ist das regelmäßige und automatisierte Harvesting von Ressourcen mit anschließender Indexierung. Auf Ubuntu wäre es möglich, über den Kommandozeilenbefehl `cron` weitere Befehle zu bestimmten Zeiten ausführen zu lassen. Die für Kapitel 3.2 erstellten und genutzten Skripte müssten dementsprechend angepasst werden. Um nicht alle Datensätze erneut herunterladen und indexieren zu müssen, würde ein Abgleich der bereits indexierten und der zu harvestenden Identifier erfolgen.

Einhergehend mit der Automatisierung ginge das Hinzufügen weiterer Ressourcen in den LISard-Index. Auf lange Sicht wäre es denkbar, mit dem `yaz`-Tool LISard an andere Kataloge über Z39.50- oder SRU-Schnittstellen anzubinden und auch von dort Metadaten zu harvesten. Weiterhin wäre die Integration ganzer Zeitschriftenjahrgänge wie z.B. von Library Hi Tech oder vom Journal of the American Society for Information Science and

Technology erstrebenswert.

Anhand neuer Datensätze aus den regelmäßig geharvesteten Ressourcen ließe sich automatisch eine Neuerscheinungsliste für die Bibliotheks- und Informationswissenschaft generieren. Zusätzlich könnte diese über RSS abonniert werden.

Damit bessere Retrievalergebnisse erzielt werden können, sollten Volltexte mitindexiert werden. Um unautorisierte Zugriffe auf Volltexte kommerziell orientierter Anbieter zu Vermeiden, wäre es nötig, den Aufruf der Daten auf einen bestimmten IP-Adressbereich einzuschränken oder mit Identitätsmanagementsystemen wie Shibboleth zu regeln.

Selbst der Harvest der Metadaten für die bestehende LISard-Kollektion muss als rechtlich ambivalent betrachtet werden. Für autorisierte Institutionen wie (andere) Bibliotheken und Archive bieten die UB-HU Berlin und DNB Schnittstellen für den (kostenpflichtigen) Austausch von Metadaten an. Der Zugriff auf die Datensätze jener Ressourcen erfolgte zwar nicht über diese Schnittstelle, allerdings können die Datensätze in mühsamer Klickarbeit von jedem Nutzer aufgerufen und heruntergeladen werden. Aufgrund der rechtlichen Bedenken kann LISard (vorerst) nicht öffentlich freigegeben werden.

Die ursprüngliche „Inspiration“ für diese Arbeit und gleichzeitige „Konkurrenz“ b2i hat Verträge mit den dort integrierten Anbietern und ist keinem rechtlichen Bedenken ausgesetzt. Dennoch konnte mit LISard demonstriert werden, dass ein Fachinformationsportal für die Bibliotheks- und Informationswissenschaften auch mit zeitgemäßer Suchmaschinenteknologie betrieben werden kann.

Suchmaschinenuntypisch schlüsselt b2i seine Trefferliste gentrennt nach Datenbanken auf. Des Weiteren bietet b2i keine Rechtschreibkorrektur bzw. Vorschläge für neue Suchen, wenn eine Anfrage aufgrund eines Fehlers keine Treffer findet. Werden Treffer gefunden, werden die Suchterme nicht in der Resultatsliste hervorgehoben; dem Nutzer wird nicht sofort klar, warum dieser Treffer für ihn relevant sein könnte. Eine Autovervollständigung bzw. automatische Vorschläge für Suchanfragen bei Eingabe der Query gibt es bei b2i ebenso nicht.

Bei der Nutzung von b2i fehlen einem generischen Nutzer Funktionen, die er bereits vom alltäglichen Suchmaschinengebrauch kennt und voraussetzt (Sadeh, 2007a; Novotny, 2004). Unter anderem sind es diese fehlenden Funktionen, die informatives Feedback für den Nutzer liefern und so eine bessere Benutzungserfahrung schaffen kann (Hearst, 2009).

LISard hingegen hat all diese Funktionen implementiert und bietet zusätzlich eine einheitliche Trefferliste über alle abgesuchten Ressourcen. Der Nutzer bekommt schon während er den ersten Buchstaben in die Suchmaske eintippt Suchvorschläge; Suchterme werden in der Resultatsliste hervorgehoben; vertippt sich der Nutzer, erhält er von LISard automatisch einen Korrekturvorschlag, sodass er dennoch seine Anfrage ausführen und Treffer erhalten kann. Die vereinte Trefferliste nimmt die Last vom Nutzer wissen zu

müssen, in welcher Datenbank sich relevante Einträge verbergen (Dietz et al., 2007).

Die Machbarkeit (und Durchführbarkeit) der Erstellung eines Fachinformationsportal für die Bibliotheks- und Informationswissenschaft mit integriertem Index und vereinter Trefferliste auf VuFind-Basis wurde anhand von LISard demonstriert:

Mit LISard wurde Schellhasen Definition eines Fachinformationsportals verwirklicht: LISard „bietet Zugänge zu fachspezifischen Publikationen“ (Zeitschriftenartikel, Konferenzbeiträge, e-Books, etc.) und „[integriert] unterschiedliche bibliographische Systeme“ (Bibliographie, Katalog, Repository, Verlagsangebot) und trägt so dazu bei „die bestehende Zersplitterung von fachlichen Nachweissystemen zu überwinden.“

## A Literaturverzeichnis

Alle Links wurden zuletzt abgerufen am: 21.12.2012

- [Antelman et al. 2006] ANTELMAN, Kristin ; LYNEMA, Emily ; PACE, Andrew K.: Toward a 21st Century Library Catalog. In: *Information Technology and Libraries* 25 (2006), Nr. 3, 128–139. <http://hdl.handle.net/10760/8177>. – ISSN 0730–9295
- [Borgman 1986] BORGMAN, Christine L.: Why are Online Catalogs Hard to Use? Lessons Learned from Information-Retrieval Studies. In: *Journal of the American Society for Information Science* 37 (1986), Nr. 6, 387–400. [http://onlinelibrary.wiley.com/doi/10.1002/\(SICI\)1097-4571\(198611\)37:6<387::AID-ASI3>3.0.CO;2-8/pdf](http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-4571(198611)37:6<387::AID-ASI3>3.0.CO;2-8/pdf). – ISSN 1532–2890
- [Deutsche Forschungsgemeinschaft 1998] DEUTSCHE FORSCHUNGSGEMEINSCHAFT: Weiterentwicklung der überregionalen Literaturversorgung / Deutsche Forschungsgemeinschaft. Version: 1998. [http://www.dfg.de/download/pdf/dfg\\_im\\_profil/reden\\_stellungnahmen/download/memo.pdf](http://www.dfg.de/download/pdf/dfg_im_profil/reden_stellungnahmen/download/memo.pdf). 1998. – Forschungsbericht
- [Dietz et al. 2007] DIETZ, Gunnar ; JUHRISCH, Martin ; KUSS MANN, Dirk ; SCHUMACHER, Frank ; STOYCHEV, Stanislav ; STRACKE, Martin: Integriertes Informationsmanagement an einer großen Universität. In: GAEDKE, Martin (Hrsg.) ; BORGEESE, Rolf (Hrsg.): *Integriertes Informationsmanagement an Hochschulen*. Karlsruhe : Universitätsverlag Karlsruhe, 2007. – ISBN 9783866441125, 57–74
- [English et al. 2002] ENGLISH, Jennifer ; HEARST, Marti ; SINHA, Rashmi ; SWEARINGEN, Kirsten ; YEE, Ka-Ping: Flexible Search and Navigation using Faceted Metadata / School of Information Management & Systems, University of Berkeley. Version: 2002. [http://oc.ac.ge/pluginfile.php/451/mod\\_resource/content/0/\\_2\\_English\\_et\\_al\\_2003\\_Flexible\\_Search\\_and\\_Navigation\\_using\\_Faceted\\_Metadata.pdf](http://oc.ac.ge/pluginfile.php/451/mod_resource/content/0/_2_English_et_al_2003_Flexible_Search_and_Navigation_using_Faceted_Metadata.pdf). Berkeley, 2002. – Forschungsbericht
- [Fox 1989] FOX, Christopher: A stop list for general text. In: *SIGIR Forum* 24 (1989), September, Nr. 1-2, 19–21. <http://dx.doi.org/10.1145/378881.378888>. – DOI 10.1145/378881.378888. – ISSN 0163–5840
- [Greifeneder 2012] GREIFENEDER, Elke: *Does it matter where we test? Online user studies in digital libraries in natural environments*, Humboldt-Universität zu Berlin, Dissertation, 2012. <http://nbn-resolving.de/urn:nbn:de:kobv:11-100203293>
- [Guba 2006] GUBA, Beate: Bibliotheken und Portale - ein Überblick. In: *GMS Medizin — Bibliothek — Information* 6 (2006), Nr. 2. <http://scholar.google.com/scholar?>

- hl=en&btnG=Search&q=intitle:Bibliotheken+und+Portale+-+ein+?berblick#  
0http://www.egms.de/static/de/journals/mbi/2006-6/mbi000033.shtml. –  
ISSN 1865-066X
- [Hänger et al. 2010] HÄNGER, Christian ; KALDENBERG, Bettina ; PFEFFER, Magnus:  
Präsentation von E-Books mit Primo. In: *Bibliotheksdienst* 44 (2010), Nr. 7, S. 711–715.  
– ISSN 0006-1972
- [Harter 1986] HARTER, Stephen P.: *Online information retrieval: concepts, principles,  
and techniques*. San Diego, CA, USA : Academic Press Professional, Inc., 1986. – ISBN  
0-12-328456-2
- [Hearst 2009] HEARST, Marti A.: Search User Interfaces. In: *Interfaces*. Cambridge :  
Cambridge University Press, 2009. – ISBN 9780521113793, Kapitel 1
- [Höchstötter u. Lewandowski 2009] HÖCHSTÖTTER, Nadine ; LEWANDOWSKI, Dirk: What  
users see – Structures in search engine results pages. In: *Information Sciences* 179  
(2009), Mai, Nr. 12, 1796–1812. <http://dx.doi.org/10.1016/j.ins.2009.01.028>. –  
DOI 10.1016/j.ins.2009.01.028. – ISSN 0020-0255
- [Hollnagel u. Woods 1983] HOLLNAGEL, Erik ; WOODS, David D.: Cognitive Systems  
Engineering: New wine in new bottles. In: *International Journal of Man-Machine  
Studies* 18 (1983), Juni, Nr. 6, 583–600. [http://www.sciencedirect.com/science/  
article/pii/S0020737383800340](http://www.sciencedirect.com/science/article/pii/S0020737383800340). – ISSN 0020-7373
- [Houser 2009] HOUSER, John: The VuFind implementation at Villanova Universi-  
ty. In: *Library Hi Tech* 27 (2009), Nr. 1, 93–105. [http://dx.doi.org/10.1108/  
07378830910942955](http://dx.doi.org/10.1108/07378830910942955). – DOI 10.1108/07378830910942955. – ISSN 0737-8831
- [Jansen et al. 1998] JANSEN, Bernard J. ; SPINK, Amanda ; BATEMAN, Judy ; SARACEVIC,  
Tefko: Real life information retrieval: a study of user queries on the Web. In: *SIGIR  
Forum* 32 (1998), April, Nr. 1, 5–17. <http://dx.doi.org/10.1145/281250.281253>. –  
DOI 10.1145/281250.281253. – ISSN 0163-5840
- [Katz 2011] KATZ, Demian: *VuFind Beyond MARC: Discovering Everything Else*. [http:  
//www.code4lib.org/conference/2011/katz](http://www.code4lib.org/conference/2011/katz). Version: 2011
- [Kinstler 2010] KINSTLER, Till: *VuFind - Ein Open Source Projekt*. [http://www.  
opus-bayern.de/bib-info/volltexte//2010/823](http://www.opus-bayern.de/bib-info/volltexte//2010/823). Version: März 2010

- [Kirchhof et al. 2006] KIRCHHOF, Anja ; GURZKI, Thorsten ; HINDERER, Henning ; VLACHAKIS, Joannis: Was ist ein Portal?: Definition und Einsatz von Unternehmensportalen / Fraunhofer Institut für Arbeitswirtschaft und Organisation. Version: 2006. [http://pub-379.bi.fraunhofer.de/Images/Was\\_ist\\_ein\\_Portal\\_tcm379-129040.pdf](http://pub-379.bi.fraunhofer.de/Images/Was_ist_ein_Portal_tcm379-129040.pdf). 2006. – Forschungsbericht
- [Krovetz 1993] KROVETZ, Robert: Viewing morphology as an inference process. In: *Proceedings of the 16th annual international ACM ...*, 1993. – ISBN 0897916050, 191–202
- [Kules et al. 2009] KULES, Bill ; CAPRA, Robert ; BANTA, Matthew ; SIERRA, Tito: What do exploratory searchers look at in a faceted search interface? In: *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*. New York, NY, USA : ACM, 2009 (JCDL '09). – ISBN 978-1-60558-322-8, 313–322
- [Langenstein u. Maylein 2009] LANGENSTEIN, Annette ; MAYLEIN, Leonhard: Relevanz-Ranking im OPAC der Universitätsbibliothek Heidelberg. In: *B.I.T. online* 12 (2009), Nr. 4, 408–413. [http://archiv.ub.uni-heidelberg.de/volltextserver/10343/1/Langenstein\\_Maylein\\_aus\\_BIT\\_4\\_09\\_kpl\\_kl.pdf](http://archiv.ub.uni-heidelberg.de/volltextserver/10343/1/Langenstein_Maylein_aus_BIT_4_09_kpl_kl.pdf). – ISSN 1435-7607
- [Lovins 1968] LOVINS, Julie B.: Development of a stemming algorithm. In: *Mechanical Translation and Computational Linguistics* 11 (1968), Nr. 1-2, 22–31. <http://journal.mercubuana.ac.id/data/MT-1968-Lovins.pdf>
- [Marchionini 1995] MARCHIONINI, Gary: *Information Seeking in Electronic Environments*. Cambridge : Cambridge University Press, 1995. – ISBN 0-521-44372-5
- [Marty u. Twidale 2005] MARTY, PF ; TWIDALE, MB: Extreme discount usability engineering. (2005). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.3702&rep=rep1&type=pdf>
- [Meyenburg 2000] MEYENBURG, Sven: Der Aufbau Virtueller Fachbibliotheken in der Bundesrepublik Deutschland. In: *Bibliotheksdienst* 34 (2000), Nr. 7/8, 1229–1235. [http://bibliotheksdienst.zlb.de/2000/2000\\_07u08\\_Informationsvermittlung01.pdf](http://bibliotheksdienst.zlb.de/2000/2000_07u08_Informationsvermittlung01.pdf). – ISSN 0006-1972
- [Morrison et al. 2007] MORRISON, By H. ; COLL, Imma S. ; ROBBIO, Antonella D. ; FOUNDER, E-lis: E-LIS : The Open Archive for Library and information Science. In: *The Charleston Advisor* 9 (2007), Nr. 1, 56–61. <http://hdl.handle.net/2429/941>. – ISSN 1525-4003



- [Novotny 2004] NOVOTNY, Eric: I don't think I click: A protocol analysis study of use of a library online catalog in the Internet age. In: *College & Research Libraries* 65 (2004), Nr. 6, 525–537. <http://crl.acrl.org/content/65/6/525.short>. – ISSN 2150–6701
- [Porter 1980] PORTER, Martin F.: An algorithm for suffix stripping. In: *Program: electronic library and information systems* 14 (1980), Nr. 3, 130–137. <http://www.lirmm.fr/~mroche/Recherche/Articles/Porter/porter.pdf>. – ISSN 1758–7301
- [Ramdeen u. Hemminger 2011] RAMDEEN, Sarah ; HEMMINGER, Bradley M.: A tale of two interfaces: How facets affect the library catalog search. In: *Journal of the American Society for Information Science and Technology* (2011), n/a–n/a. <http://dx.doi.org/10.1002/asi.21689>. – ISSN 1532–2890
- [Rösch 2001] RÖSCH, Hermann: Portale in Internet, Betrieb und Wissenschaft. In: *B.I.T. online* 4 (2001), Nr. 3, 237–248. <http://www.b-i-t-online.de/archiv/2001-03/fach1.htm>. – ISSN 1435–7607
- [Rösch u. Weisbrod 2004] RÖSCH, Hermann ; WEISBROD, Dirk: Linklisten, Subject Gateways, Virtuelle Fachbibliotheken, Bibliotheks- und Wissenschaftsportale: Typologischer Überblick und Definitionsvorschlag. In: *BIT Online* 7 (2004), Nr. 3, 177–188. <http://www.b-i-t-online.de/archiv/2004-03/fach1.htm>. – ISSN 1435–7607
- [Sadeh 2007a] SADEH, Tamar: Time for a change: new approaches for a new generation of library users. In: *New Library World* 108 (2007), Nr. 7/8, 307–316. <http://dx.doi.org/10.1108/03074800710763608>. – DOI 10.1108/03074800710763608. – ISSN 0307–4803
- [Sadeh 2007b] SADEH, Tamar: User-Centric Solutions for Scholarly Research in the Library. In: *Liber Quarterly* 17 (2007), Nr. 3-4. – ISSN 1435–5205
- [Salton et al. 1975] SALTON, Gerald ; WONG, Andrew ; YANG, Chung S.: A vector space model for automatic indexing. In: *Commun. ACM* 18 (1975), November, Nr. 11, 613–620. <http://dx.doi.org/10.1145/361219.361220>. – DOI 10.1145/361219.361220. – ISSN 0001–0782
- [Salton 1989] SALTON, Gerard: *Automatic text processing : the transformation, analysis, and retrieval of information by computer*. Reprinted. Reading, Mass. [u.a.] : Addison-Wesley, 1989
- [Salton et al. 1983] SALTON, Gerard ; FOX, Edward A. ; WU, Harry: Extended Boolean information retrieval. In: *Commun. ACM* 26 (1983), November, Nr. 11, 1022–1036.

- <http://dx.doi.org/10.1145/182.358466>. – DOI 10.1145/182.358466. – ISSN 0001–0782
- [Schellhase 2008] SCHELLHASE, Jörg: Systeme und Dienste zur Unterstützung wissenschaftlicher Rechercheprozesse. In: *Recherche Wissenschaftlicher Publikationen*. 1. Lohmar : Eul, 2008. – ISBN 978–3899366914, Kapitel 4, S. 131 – 191
- [Schmid 2009] SCHMID, Helmut: Tokenizing and part-of-speech tagging. Version: 2009. <http://www.coli.uni-saarland.de/~schulte/Teaching/ESSLI-06/Referenzen/Tokenisation/schmid-hsk-tok.pdf>. In: LÜDELING, Anne (Hrsg.) ; KYTÖ, Merja (Hrsg.) ; MCENERY, Tony (Hrsg.): *Corpus Linguistics: An international Handbook*. Berlin : Walter de Gruyter, 2009. – ISBN 978–3110207330, 527–551
- [Schmitt u. Stehle 2010] SCHMITT, Jörg ; STEHLE, Marcel: *Der OPAC aus dem Baukasten: Realisierung eines Katalog 2.0 unter Einbeziehung der Community*. <http://opus.haw-hamburg.de/volltexte/2011/1143/>. Version: 2010
- [Seeborn 2011] SEEBOHN, Joachim: Farben. Version: 2011. [http://dx.doi.org/10.1007/978-3-8349-6433-5\\_6](http://dx.doi.org/10.1007/978-3-8349-6433-5_6). In: *Gabler Kompaktlexikon Werbung*. Gabler, 2011. – DOI 10.1007/978–3–8349–6433–5\_6. – ISBN 978–3–8349–0148–4
- [Seefeldt u. Syré 2007] SEEFELDT, Jürgen ; SYRÉ, Ludger: *Portale zu Vergangenheit und Zukunft : Bibliotheken in Deutschland*. 3., überar. Hildesheim [u.a.] : Hildesheim [u.a.] : Olms, 2007. – ISBN 978–3–487–13347–8
- [Stock 2008] STOCK, Wolfgang G.: *Wissensrepräsentation : Informationen auswerten und bereitstellen*. München : München : Oldenbourg, 2008 978–3–486–58439–4
- [Tam et al. 2009] TAM, Winnie ; COX, Andrew M. ; BUSSEY, A.: Student user preferences for features of next-generation OPACs: a case study of University of Sheffield international students. In: *Program: electronic library and information systems* 43 (2009), Nr. 4, S. 349–374. <http://dx.doi.org/http://dx.doi.org/10.1108/00330330910998020>. – DOI <http://dx.doi.org/10.1108/00330330910998020>. – ISSN 1758–7301
- [Taylor u. Joudrey 2009] TAYLOR, Arlene G. ; JOUDREY, Daniel N.: Systems and System Design. In: *The Organization of Information*. 3. ed., 1. Westport : Libraries Unlimited, 2009. – ISBN 978–1–59158–586–2, Kapitel 6, S. 159 – 198
- [Wilbur u. Sirotkin 1992] WILBUR, W. J. ; SIROTKIN, K.: The automatic identification of stop words. In: *Journal of Information Science* 18 (1992), Januar, Nr. 1, 45–55. <http://dx.doi.org/10.1177/016555159201800106>. – DOI 10.1177/016555159201800106. – ISSN 0165–5515

- [Willett 2006] WILLETT, Peter: The Porter stemming algorithm: then and now. In: *Program: electronic library and information systems* 40 (2006), 219–223. <http://www.emeraldinsight.com/journals.htm?articleid=1563486&show=abstract>. – ISSN 1758–7301
- [Wissenschaftsrat 2012] WISSENSCHAFTSRAT: Empfehlungen zur Weiterentwicklung der wissenschaftlichen Informationsinfrastruktur in Deutschland bis 2020 / Wissenschaftsrat. Version: Juli 2012. <http://www.wissenschaftsrat.de/download/archiv/2359-12.pdf>. 2012. – Forschungsbericht
- [Xu u. Croft 1998] XU, Jinxi ; CROFT, W. B.: Corpus-based stemming using cooccurrence of word variants. In: *ACM Transactions on Information Systems* 16 (1998), Januar, Nr. 1, 61–81. <http://dx.doi.org/10.1145/267954.267957>. – DOI 10.1145/267954.267957. – ISSN 1046–8188

## B Andere Quellen

Alle Links wurden zuletzt abgerufen am: 21.12.2012

- [Deutsche Nationalbibliothek 2012] DEUTSCHE NATIONALBIBLIOTHEK: *Datenshop*. <http://www.dnb.de/DE/Header/Hilfe/datenshop.html>. Version: Dezember 2012
- [DNBG 2006] DNBG: *Gesetz über die Deutsche Nationalbibliothek*. <http://www.gesetze-im-internet.de/dnbg/BJNR133800006.html>. Version: 2006
- [Drößler 2012] DRÖSSLER, Stefan: *Neues Rechercheportal VuFind kommt 2013*. <http://blog.ub.uni-kassel.de/ub/2012/05/22/rechercheportal-vufind/>. Version: 2012
- [ExLibris 2009] EXLIBRIS: *Primo Search*. <http://www.exlibrisgroup.com/files/Products/Primo/PrimoSearchWhitePaperDec2009.pdf>. Version: 2009
- [ExLibris 2012] EXLIBRIS: *Primo Overview*. <http://www.exlibrisgroup.com/category/PrimoOverview>. Version: 2012
- [Hostetter 2010] HOSTETTER, Chris: *What's a "DisMax"?* <http://searchhub.org/2010/05/23/whats-a-dismax/>. Version: Mai 2010
- [Library of Congress 2010] LIBRARY OF CONGRESS: *008 - Fixed-Length Data Elements-General Information (NR)*. <http://www.loc.gov/marc/bibliographic/bd008.html>. Version: 2010
- [Library of Congress 2012] LIBRARY OF CONGRESS: *041 - Language Code (R)*. <http://www.loc.gov/marc/bibliographic/bd041.html>. Version: September 2012
- [Lucene-Wiki 2012] LUCENE-WIKI: *Scoring*. [http://lucene.apache.org/core/old\\_versioned\\_docs/versions/2\\_9\\_1/scoring.html](http://lucene.apache.org/core/old_versioned_docs/versions/2_9_1/scoring.html). Version: Juni 2012
- [Naval Postgraduate School 2012] NAVAL POSTGRADUATE SCHOOL: *How to index DSpace with VuFind*. [http://vufind.org/wiki/importing\\_records:how\\_to\\_index\\_dspace\\_with\\_vufind](http://vufind.org/wiki/importing_records:how_to_index_dspace_with_vufind). Version: Mai 2012
- [Porter 2006] PORTER, Martin F.: *Stemming algorithms for various European languages*. <http://snowball.tartarus.org/texts/stemmersoverview.html>. Version: 2006
- [Savoy 2005a] SAVOY, Jacques: *English Stopwordlist*. <http://members.unine.ch/jacques.savoy/clef/englishST.txt>. Version: 2005
- [Savoy 2005b] SAVOY, Jacques: *German Stopwordlist*. <http://members.unine.ch/jacques.savoy/clef/germanST.txt>. Version: 2005

- [Savoy 2005c] SAVOY, Jacques: *IR Multilingual Resources at UniNE*. <http://members.unine.ch/jacques.savoy/clef/>. Version: 2005
- [SOLR-Wiki 2012a] SOLR-WIKI: *AnalyzersTokenizersTokenFilters*. <http://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters>. Version: 2012
- [SOLR-Wiki 2012b] SOLR-WIKI: *Language-Analysis*. <http://wiki.apache.org/solr/LanguageAnalysis#German>. Version: September 2012
- [Solrmarc-Wiki 2010] SOLRMARC-WIKI: *IndexProperties: Details on the index.properties file*. <http://code.google.com/p/solrmarc/wiki/IndexProperties>. Version: Oktober 2010
- [SpringerLink 2012a] SPRINGERLINK: *MARC Records for Springer eBooks & SpringerProtocols*. <http://www.springer.com/?referer=springer.com&SGWID=1-148802-3020-0-0>. Version: Dezember 2012
- [SpringerLink 2012b] SPRINGERLINK: *Willkommen bei SpringerLink*. <http://www.springer.com/help?SGWID=1-114-12-349801-0>. Version: 2012
- [VuFind-Dokumentation 2012a] VUFIND-DOKUMENTATION: *Autocomplete*. <http://vufind.org/wiki/autocomplete>. Version: September 2012
- [VuFind-Dokumentation 2012b] VUFIND-DOKUMENTATION: *Book Previews*. [http://vufind.org/wiki/book\\_previews](http://vufind.org/wiki/book_previews). Version: August 2012
- [VuFind-Dokumentation 2012c] VUFIND-DOKUMENTATION: *Importing Records*. [http://vufind.org/wiki/importing\\_records](http://vufind.org/wiki/importing_records). Version: Oktober 2012
- [VuFind-Dokumentation 2012d] VUFIND-DOKUMENTATION: *Search Customization*. [http://vufind.org/wiki/searches\\_customizing\\_tuning\\_adding](http://vufind.org/wiki/searches_customizing_tuning_adding). Version: August 2012
- [VuFind-Dokumentation 2012e] VUFIND-DOKUMENTATION: *Use of External Content*. [http://vufind.org/wiki/use\\_of\\_external\\_content](http://vufind.org/wiki/use_of_external_content). Version: August 2012
- [VuFind-Dokumentation 2012f] VUFIND-DOKUMENTATION: *Working with Facets*. [http://vufind.org/wiki/working\\_with\\_facets](http://vufind.org/wiki/working_with_facets). Version: August 2012
- [Wikibooks 2011] WIKIBOOKS: *Webseitenentwicklung: Grafikdesign: Farben*. [http://de.wikibooks.org/w/index.php?title=Websiteentwicklung:\\_Grafikdesign:\\_Farben&oldid=565433](http://de.wikibooks.org/w/index.php?title=Websiteentwicklung:_Grafikdesign:_Farben&oldid=565433). Version: Januar 2011

[Zierold 2011] ZIEROLD, Marius: *Der neue Onlinekatalog der Genderbibliothek*. <http://marius300482.wordpress.com/2011/09/23/der-neue-onlinekatalog-der-genderbibliothek-gendervu/>. Version: September 2011

# C Abbildungen

The screenshot shows the b2i search interface. The search query is 'informationssystem'. The results are categorized by database, with three specific database result lists highlighted by red boxes:

- Treffer Katalog St. Galler Zentrum für das Buch, Kantonsbibliothek Vadana St. Gallen (490)**: This list contains 5 results, including 'Statistisches Informationssystem des Bundes' and 'Das Internationale Informationssystem für Agrarwissenschaft und -technik (AGRIS)'. Navigation links 1 through 9 are visible.
- Treffer Fach-OPAC SUB Göttingen (364)**: This list contains 5 results, including 'META-Referenzarchitektur für ein statistisches Informationssystem' and 'Die Datenbank im Informationssystem'. Navigation links 1 through 9 are visible.
- Treffer Fachdatenbank Buchwissenschaft, SUB Göttingen (BBB) (29)**: This list contains 1 result, 'Peter Suhrkamp 1891-1959 : 3777 Peter Suhrkamp 1891-1959. Vom Bauernsohn aus Kirchhatten zum Frankfurter Verleger. Eine Ausstellung zum 100. Geburtstag am 28. März; Oldenburg, Bibliotheks- und Informationssystem der Universität Oldenburg. Red. und Bearb.: Eva Schreiber. Hrsg. von Dirk Grathoff. - Oldenburg: Bibliotheksund Informationssystem der Universität Oldenburg 1991. 72 S., Ill. (Darin: Bibliographie der Schriften von Peter Suhrkamp S. 53-72)'. Navigation links 1 through 6 are visible.

The left sidebar provides filters for Thema, Sprache, Publikationsart/-form, and Autor / Institution. The bottom section lists various databases available in the system, such as 'Katalog St. Galler Zentrum für das Buch' and 'Fachdatenbank Buchwissenschaft'.

Abbildung C.1: Resultatsliste bei b2i nach einzelnen Datenbanken aufgeschlüsselt

**Datenbankbereich**

[Datenbank-Auswahl ändern](#)  
[ändern](#)

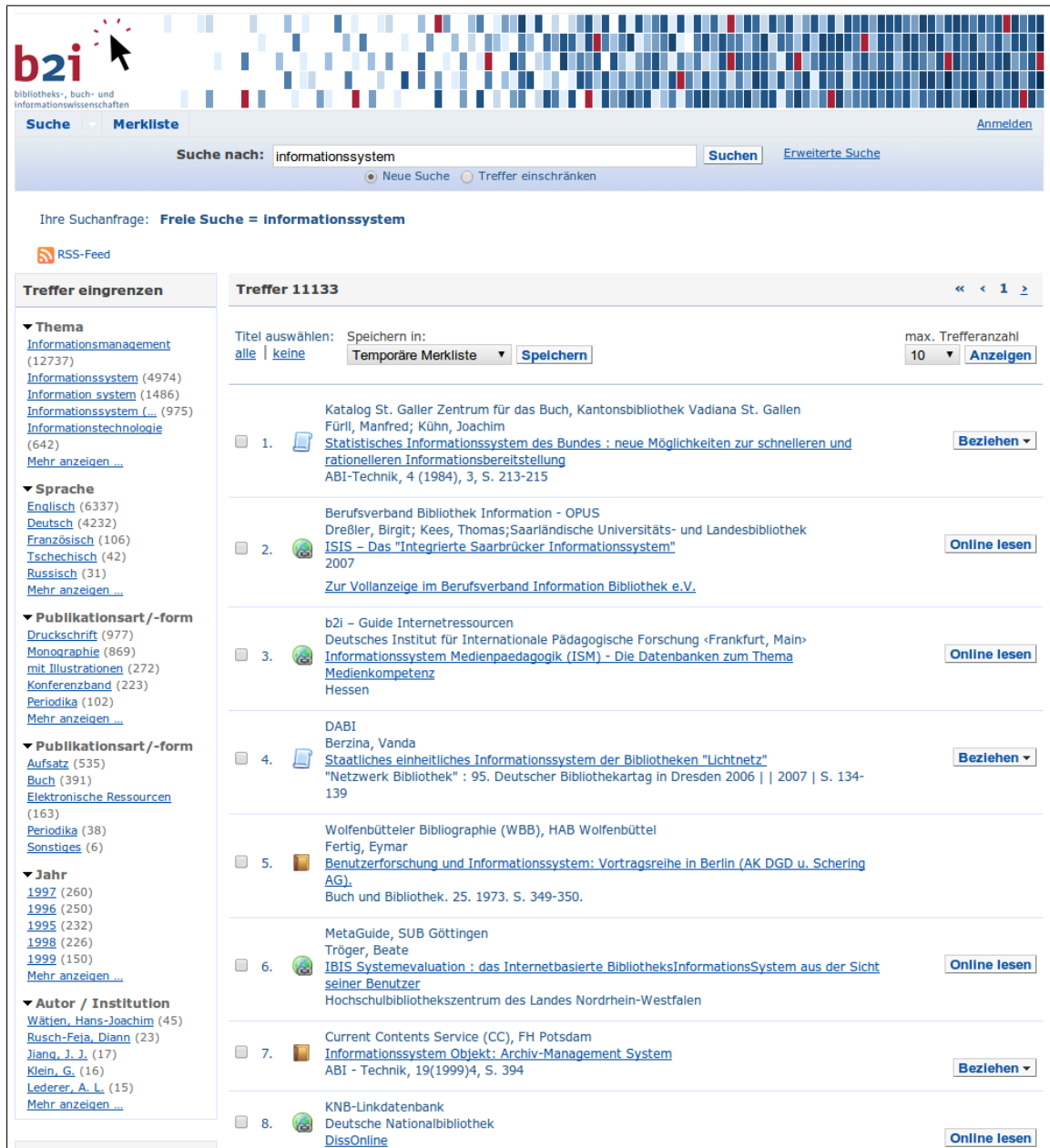
**Datenbanken (30)**

BSB-Ink online (0)  
[Katalog St. Galler Zentrum für das Buch, Kantonsbibliothek Vadiana St. Gallen](#) (490)  
 Datenbank Buch und Papier, SUB Göttingen (0)  
[Fach-OPAC SUB Göttingen](#) (364)  
 BSB – Forschungsdoku.  
 Handschriften (0)  
[Fachdatenbank Buchwissenschaft, SUB Göttingen \(BBB\)](#) (29)  
[KNB Bibliotheksportal](#) (30)  
 BBGOe, Buchwissenschaft  
 Erlangen (0)  
 BSB – Codices iconographici (0)  
 Katalog LFSt ÖBIB (0)  
[Wolfenbütteler Bibliographie \(WBB\), HAB Wolfenbüttel](#) (7)  
 BSB – Einblattdrucke der frühen Neuzeit (0)  
[DigiZeitschriften, BBl-Fachausschnitt](#) (11)  
[Berufsverband Bibliothek Information - OPUS](#) (6)  
 BSB – Autogr. und Nachlässe (0)  
[b2i – Guide Internetressourcen](#) (46)  
[DABI](#) (35)  
[MetaGuide, SUB Göttingen](#) (1)  
[Current Contents Service \(CC\), FH Potsdam](#) (9)  
[KNB-Linkdatenbank](#) (2)  
 E-LIS (0)  
[OLC-SSG-Fachausschnitt, SUB Göttingen](#) (31)  
[INFODATA-eDepot, FH Potsdam](#) (95)  
[INFODATA, FH Potsdam \(eingeschr. Zugang\)](#) (7379)  
 DNB - Buchhändlerische Geschäftsrunderheben des Deutschen Buch- und Schriftmuseums (0)  
[Katalog der Gutenberg-Bibliothek](#) (7)  
[BSB - Fachopac b2i](#) (1017)  
[DOBI](#) (1574)  
 Verzeichnis der Drucke des 17. Jahrhunderts (0)  
 Verzeichnis der Drucke des 16. Jahrhunderts (0)

[Alle Treffer in einer Liste anzeigen](#)

Abbildung C.2: Facette bei b2i, um Trefferlisten zusammenzufügen





The screenshot shows the b2i search interface. At the top, the b2i logo is on the left, and a search bar contains the text 'Informationssystem'. Below the search bar, there are options for 'Neue Suche' and 'Treffer einschränken'. The search results are displayed in a list format with filters on the left and a main results area on the right.

**Suche nach:** Informationssystem

**Ihre Suchanfrage:** Freie Suche = Informationssystem

**Treffer eingrenzen**

- Thema**
  - Informationsmanagement (12737)
  - Informationssystem (4974)
  - Information system (1486)
  - Informationssystem (...) (975)
  - Informationstechnologie (642)
  - Mehr anzeigen ...
- Sprache**
  - Englisch (6337)
  - Deutsch (4232)
  - Französisch (106)
  - Tschechisch (42)
  - Russisch (31)
  - Mehr anzeigen ...
- Publikationsart/-form**
  - Druckschrift (977)
  - Monographie (869)
  - mit Illustrationen (272)
  - Konferenzband (223)
  - Periodika (102)
  - Mehr anzeigen ...
- Publikationsart/-form**
  - Aufsatz (535)
  - Buch (391)
  - Elektronische Ressourcen (163)
  - Periodika (38)
  - Sonstiges (6)
- Jahr**
  - 1997 (260)
  - 1996 (250)
  - 1995 (232)
  - 1998 (226)
  - 1999 (150)
  - Mehr anzeigen ...
- Autor / Institution**
  - Wäten, Hans-Joachim (45)
  - Rusch-Feia, Diann (23)
  - Jiang, J. J. (17)
  - Klein, G. (16)
  - Lederer, A. L. (15)
  - Mehr anzeigen ...

**Treffer 11133**

Titel auswählen: ☐ alle ☐ keine   max. Trefferanzahl: 10

- ☐ 1. Katalog St. Galler Zentrum für das Buch, Kantonsbibliothek Vadana St. Gallen  
Fürl, Manfred; Kühn, Joachim  
[Statistisches Informationssystem des Bundes : neue Möglichkeiten zur schnelleren und rationelleren Informationsbereitstellung](#)  
ABI-Technik, 4 (1984), 3, S. 213-215
- ☐ 2. Berufsverband Bibliothek Information - OPUS  
Dreßler, Birgit; Kees, Thomas;Saarländische Universitäts- und Landesbibliothek  
[ISIS – Das "Integrierte Saarbrücker Informationssystem"](#)  
2007   
[Zur Vollanzeige im Berufsverband Information Bibliothek e.V.](#)
- ☐ 3. b2i – Guide Internetressourcen  
Deutsches Institut für Internationale Pädagogische Forschung <Frankfurt, Main>  
[Informationssystem Medienpädagogik \(ISM\) - Die Datenbanken zum Thema Medienkompetenz](#)  
Hessen
- ☐ 4. DABI  
Berzina, Vanda  
[Staatliches einheitliches Informationssystem der Bibliotheken "Lichtnetz"](#)  
"Netzwerk Bibliothek" : 95. Deutscher Bibliothekartag in Dresden 2006 | | 2007 | S. 134-139
- ☐ 5. Wolfenbütteler Bibliographie (WBB), HAB Wolfenbüttel  
Fertig, Eymar  
[Benutzerforschung und Informationssystem: Vortragsreihe in Berlin \(AK DGD u. Schering AG\).](#)  
Buch und Bibliothek. 25. 1973. S. 349-350.
- ☐ 6. MetaGuide, SUB Göttingen  
Tröger, Beate  
[IBIS Systemevaluation : das Internetbasierte BibliotheksInformationssystem aus der Sicht seiner Benutzer](#)  
Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen
- ☐ 7. Current Contents Service (CC), FH Potsdam  
[Informationssystem Objekt: Archiv-Management System](#)  
ABI - Technik, 19(1999)4, S. 394
- ☐ 8. KNB-Linkdatenbank  
Deutsche Nationalbibliothek  
[DissOnline](#)

Abbildung C.3: Zusammengeführte Resultatsliste bei b2i

The screenshot shows the vufind search interface. At the top, the logo 'vufind' is displayed with the tagline 'The library OPAC meets Web 2.0!'. A search bar contains the query '\*:\*' and buttons for 'Alle Felder', 'Suchen', and 'Erweitert'. The language is set to 'Deutsch'. Below the search bar, the results are sorted by 'Relevanz' and show 'Treffer 1 - 20 von 4844 für Suche: '\*:\*', Suchdauer: 2.51s'. The results list includes:

1. **Arabic name authority in the online environment : options and implications** von [Speirs, Plettner, Martha](#) Veröffentlicht 2003. [Volltext](#) [Journal Article \(Print/Paginated\)](#)
2. **Quality of Biomedical Libraries Webpages : a survey** von [Spina, Barbara](#) Veröffentlicht 2005. [Volltext](#) [Conference Poster](#)
3. **Quality of academic libraries - funding bodies, librarians and users perspective: a common project of Polish research libraries on comparable measures** von [Derfert-Wolf, Lidia](#) Veröffentlicht 2005. [Volltext](#) [Conference Paper](#)
4. **Library consortia model for country wide access of electronic journals and databases** von [Francis, A. T.](#) Veröffentlicht 2005. [Volltext](#) [Conference Paper](#)
5. **Fundamental methodologies and tools for the employment of webometric analyses : a discussion and proposal for improving the foundation of webometrics** von [Fuqi, Liv Danman](#) Veröffentlicht 2001. [Volltext](#) [Thesis](#)
6. **Open access : basics and benefits** von [Morrison, Heather](#) Veröffentlicht 2004. [Volltext](#)

On the right side, there are filters for 'Institution' (E-LIS (4844)), 'Format' (Journal Article (Print/Paginated) (1194), Conference Paper (1158), Presentation (684), Journal Article (Online/Unpaginated) (380), Preprint (293), mehr ...), 'Verfasser' (Bauer, Bruno (91), Sridhar, M. S. (72), Oberhauser, Otto (62), Raman Nair, R. (62), Holley, Rose (61), mehr ...), 'Sprache' (English (4118), German (731)), and 'Erscheinungsjahr' (Von: Bis: with a slider and a 'Wechseln' button).

Abbildung C.4: Trefferliste auf LISard nach der Indexierung der E-LIS-Datensätze

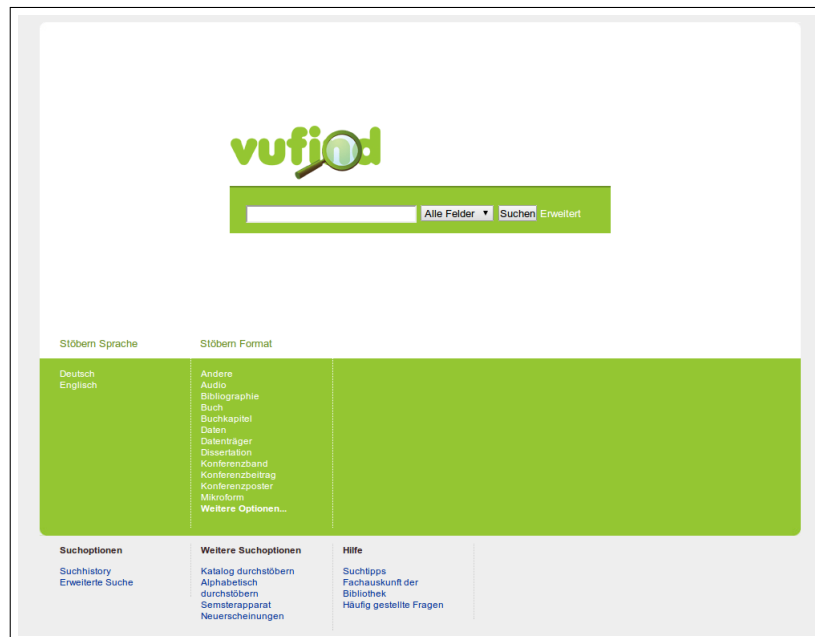


Abbildung C.5: VuFind-Startseite mit dem default-Theme

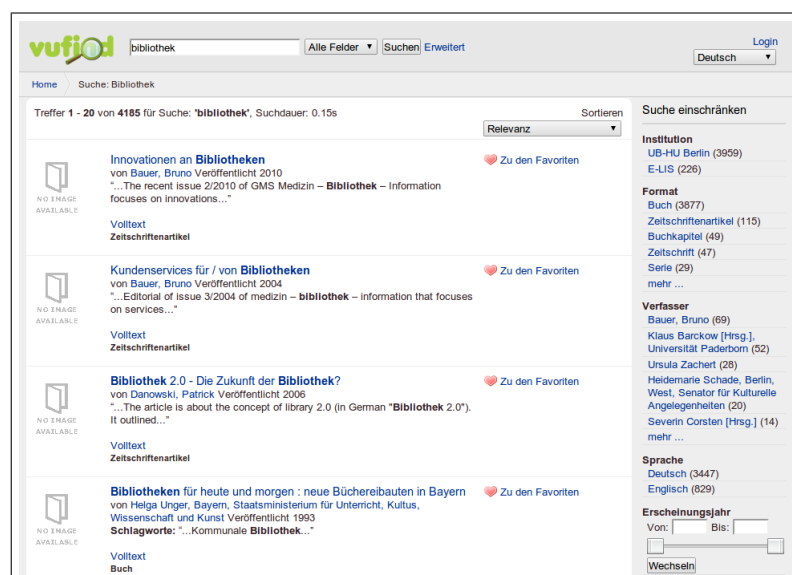


Abbildung C.6: Resultatsliste von VuFind mit dem default-Theme

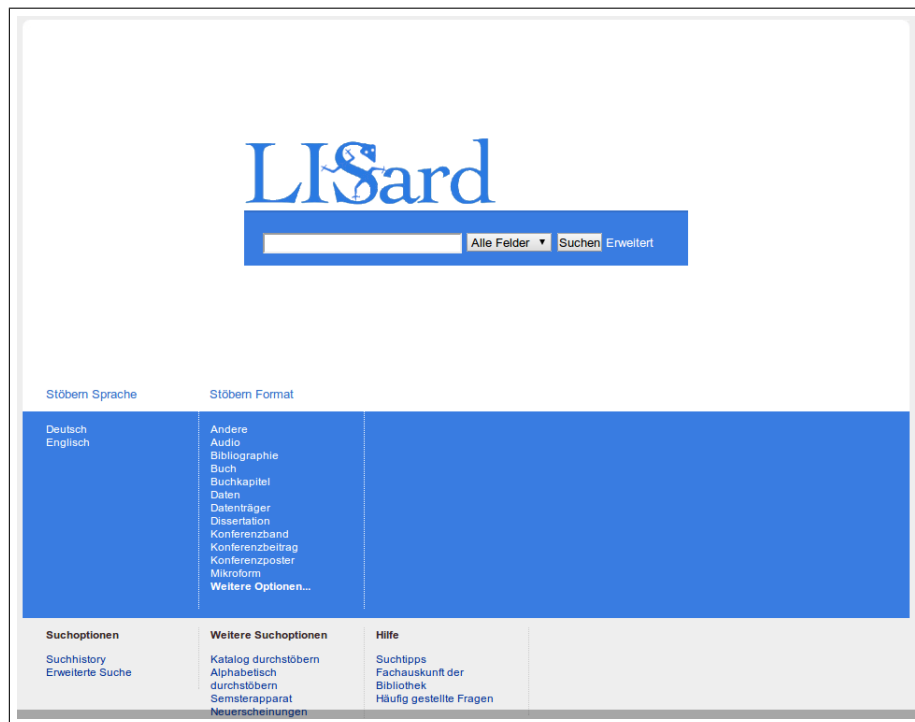


Abbildung C.7: VuFind-Startseite mit angepasstem LISard-Theme

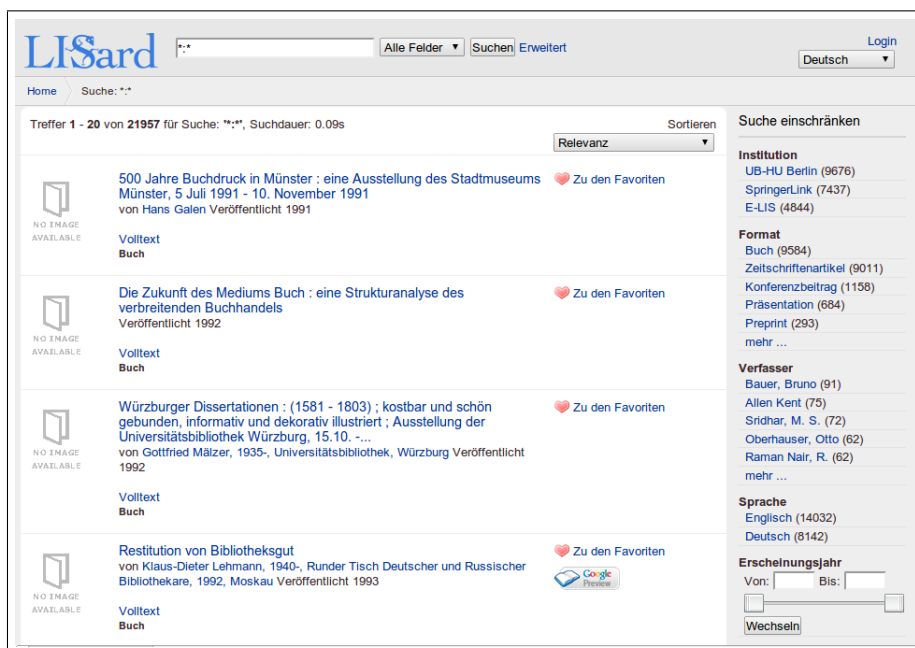


Abbildung C.8: Resultatsliste in VuFind mit angepassten LISard-Theme

<p><b>Suche einschränken</b></p> <p><b>Institution</b>  <a href="#">SpringerLink</a> (16939)  <a href="#">UB-HU Berlin</a> (9676)  <a href="#">E-LIS</a> (4844)  <a href="#">DNB</a> (89)</p> <p><b>Format</b>  <a href="#">Buch</a> (9634)  <a href="#">e-Book</a> (9506)  <a href="#">Zeitschriftenartikel</a> (9011)  <a href="#">Konferenzbeitrag</a> (1158)  <a href="#">Präsentation</a> (684)  <a href="#">mehr ...</a></p> <p><b>Signatur</b>  <a href="#">Q - Science</a> (6094)  <a href="#">T - Technology</a> (2684)  <a href="#">H - Social Science</a> (285)  <a href="#">L - Education</a> (65)  <a href="#">P - Language and Literature</a> (44)  <a href="#">mehr ...</a></p> <p><b>Verfasser</b>  <a href="#">Bauer, Bruno</a> (91)  <a href="#">Allen Kent</a> (75)  <a href="#">Sridhar, M. S.</a> (72)  <a href="#">Oberhauser, Otto</a> (62)  <a href="#">Raman Nair, R.</a> (62)  <a href="#">mehr ...</a></p> <p><b>Sprache</b>  <a href="#">Englisch</a> (14053)  <a href="#">Deutsch</a> (8219)</p> <p><b>Genre</b>  <a href="#">Zeitschrift</a> (13)  <a href="#">Kongress</a> (6)  <a href="#">Verzeichnis</a> (3)  <a href="#">Adressbuch</a> (2)  <a href="#">Bericht</a> (2)  <a href="#">mehr ...</a></p> <p><b>Ära</b>  <a href="#">Geschichte</a> (2)  <a href="#">Geschichte 1558-1579</a> (1)  <a href="#">Geschichte 1887-1940</a> (1)</p> <p><b>Region</b>  <a href="#">Deutschland</a> (6)  <a href="#">Österreich</a> (2)  <a href="#">Schweiz</a> (2)  <a href="#">Bayern</a> (1)  <a href="#">Berlin</a> (1)  <a href="#">mehr ...</a></p> <p><b>Erscheinungsjahr</b>  Von: <input type="text"/> Bis: <input type="text"/>  <input type="text"/> <input type="text"/>  <input type="button" value="Wechseln"/></p>	<p><b>Suche einschränken</b></p> <p><b>Ressource</b>  <a href="#">UB-HU Berlin</a> (9676)  <a href="#">SpringerLink</a> (7437)  <a href="#">E-LIS</a> (4844)  <a href="#">DNB</a> (73)</p> <p><b>Format</b>  <a href="#">Buch</a> (9619)  <a href="#">Zeitschriftenartikel</a> (9011)  <a href="#">Konferenzbeitrag</a> (1158)  <a href="#">Präsentation</a> (684)  <a href="#">Preprint</a> (293)  <a href="#">mehr ...</a></p> <p><b>Verfasser</b>  <a href="#">Bauer, Bruno</a> (91)  <a href="#">Allen Kent</a> (75)  <a href="#">Sridhar, M. S.</a> (72)  <a href="#">Oberhauser, Otto</a> (62)  <a href="#">Raman Nair, R.</a> (62)  <a href="#">mehr ...</a></p> <p><b>Sprache</b>  <a href="#">Englisch</a> (14050)  <a href="#">Deutsch</a> (8205)</p> <p><b>Schlagwort</b>  <a href="#">information</a> (9925)  <a href="#">scienc</a> (5659)  <a href="#">retrieval</a> (4319)  <a href="#">library</a> (4184)  <a href="#">storag</a> (4140)  <a href="#">mehr ...</a></p> <p><b>Erscheinungsjahr</b>  Von: <input type="text"/> Bis: <input type="text"/>  <input type="text"/> <input type="text"/>  <input type="button" value="Wechseln"/></p>
--	--

Abbildung C.9: Facettenleiste mit Standardeinstellungen (links), Facettenleiste für LISard angepasst (rechts)



Abbildung C.10: Resultatsliste in LISard mit angezeigter Verfügbarkeit und Standort der Treffer



Abbildung C.11: Trefferanzeige in LISard mit angezeigter Verfügbarkeit und Standort des Treffers

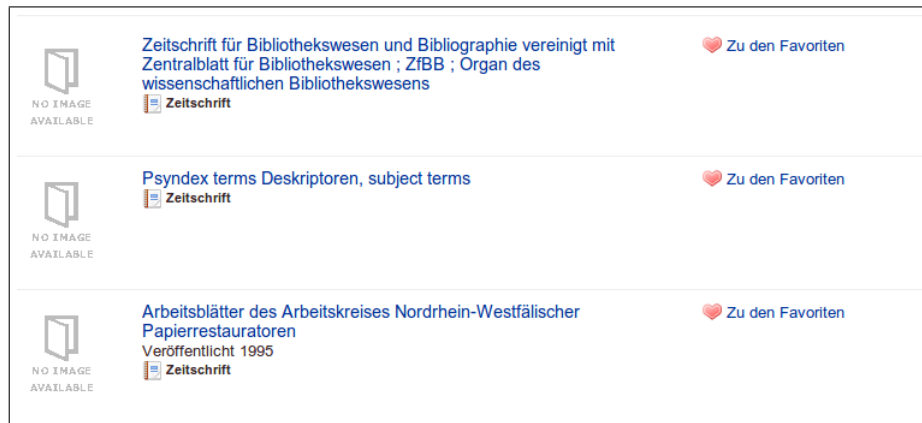


Abbildung C.12: Resultatsliste in LISard ohne angezeigter Verfügbarkeit und Standort der Treffer

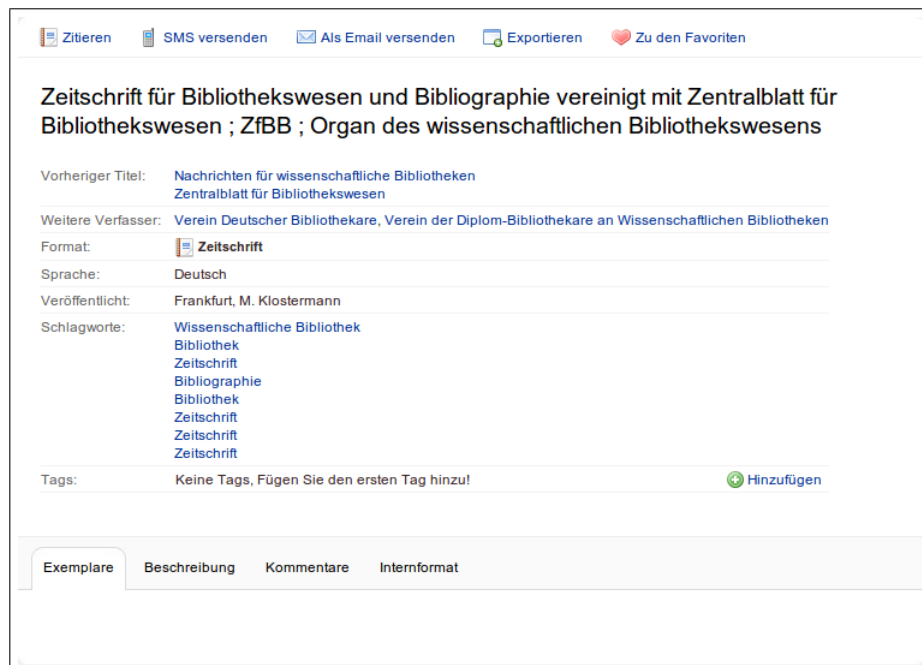


Abbildung C.13: Trefferanzeige in LISard ohne angezeigter Verfügbarkeit und Standort des Treffers

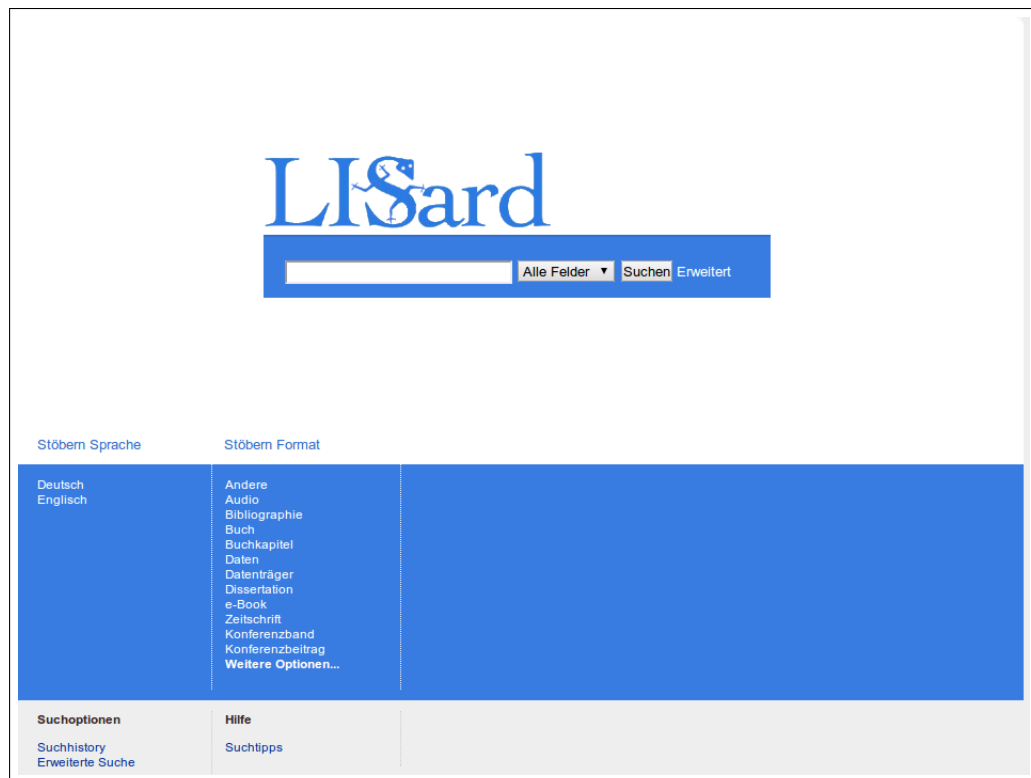


Abbildung C.14: LISard-Startseite ohne die Funktionen: FAQ, Katalog durchstöbern, alphabetisch Stöbern, Semesterapparat und Frag einen Bibliothekar

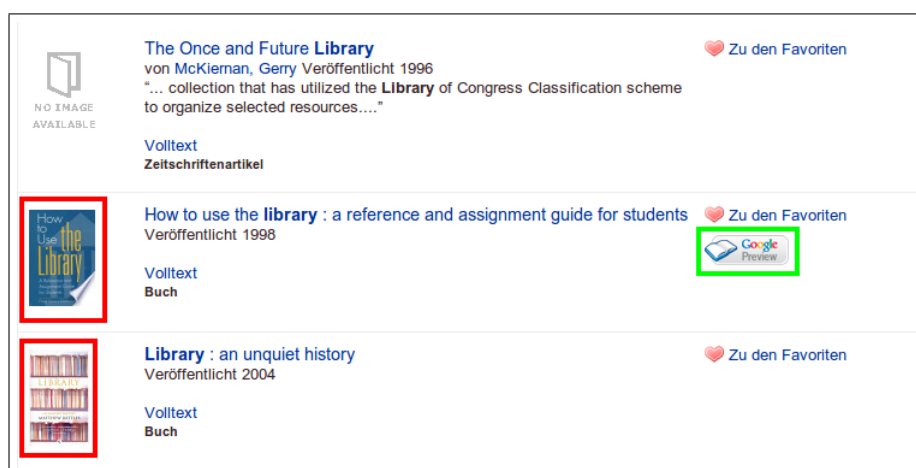


Abbildung C.15: Trefferanzeige mit Kataloganreicherung in LISard: Geladene Cover (rot), Link zu Google Books (grün)





Abbildung C.16: Rechtschreibfehlerkorrektur in LISard

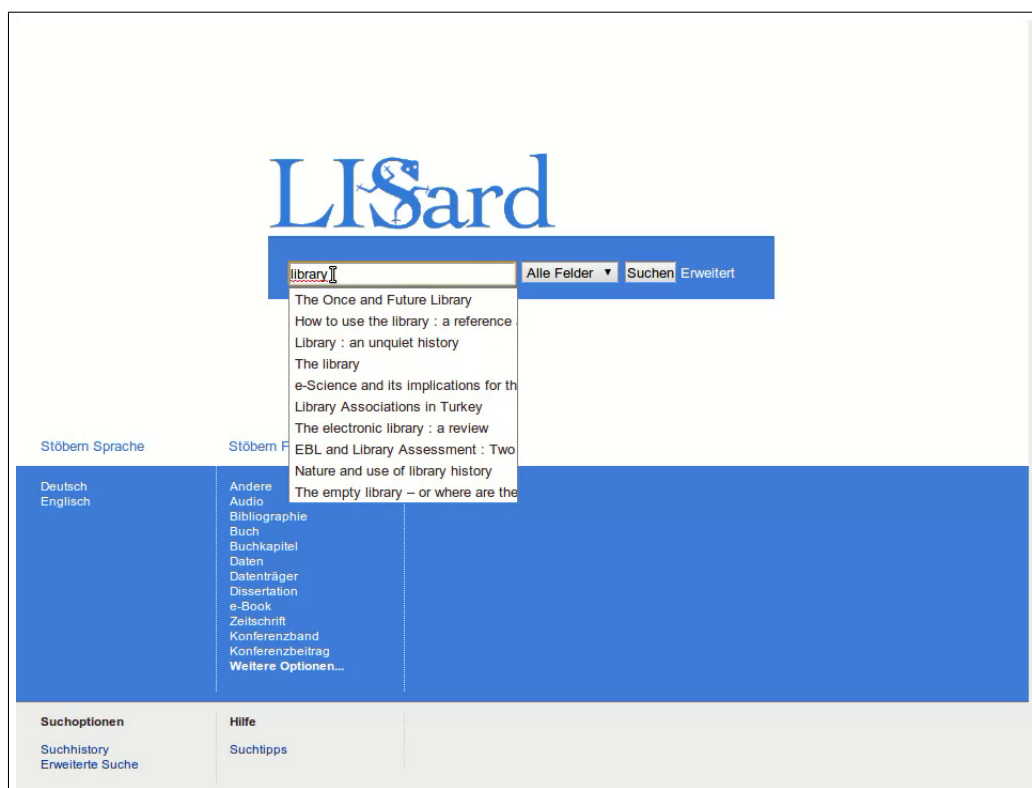


Abbildung C.17: Autovervollständigung von Suchanfragen in LISard

## D Listings

```

1 path = /vufind
2 url = http://localhost/vufind/
3 local = /usr/local/vufind/web
4 email = oliverpohl@ibi.hu-berlin.de
5 title = "LISard"
6 theme = blueprint
7 language = de
8 locale = de_DE

```

Listing 1: Angepasste /usr/local/vufind/web/conf/config.ini

```

1 [E-LIS]
2 url = http://eprints.rclis.org/dspace-oai/request
3 metadataPrefix = oai_dc
4 idSearch[] = "^oai:.eprints.rclis.org/"
5 idReplace[] = "ir-"
6 idSearch[] = "/\\/"
7 idReplace[] = "-"
8 injectId = "identifier"
9 injectDate = "datestamp"
10 harvestedIdLog = harvest-elis.log

```

Listing 2: Angepasste /usr/local/vufind/harvest/oai.ini für den Harvest von E-LIS

```

1 #!/bin/bash
2
3 for file in *xml; do
4     echo "$file";
5     EnDe='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "//dc:
        language='en' or //dc:language='de'" "$file"';
6     num_langs='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "
        count(//dc:language)" "$file"';
7     if [ "$EnDe" = "true" ] && [ $num_langs -eq 1 ]; then
8         mv "$file" "Useful/$file";
9     fi;
10    if [ "$EnDe" != "true" ]; then
11        mv "$file" "NotUseful/$file";
12    fi;
13 done

```

Listing 3: Only-DeEn-oai.sh: Shell-Skript, um nicht deutsch- oder englischsprachige Datensätze aus der E-LIS-Kollektion entfernt

```

1  #!/bin/bash
2
3  for file in *xml; do
4
5      num_lang='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "//dc:
        language" -n "$file" | wc -l';
6      i=1;
7
8      while [ $i -le $num_lang ]; do
9
10         num_lang='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "//dc:
            language" -n "$file" | wc -l';
11         lang='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "//dc:language
            [$i]" -n "$file"';
12         if [ "$lang" == "de" ] || [ "$lang" == "en" ]; then
13             echo "$file right $lang $i/$num_lang - do nothing";
14             i=$((i+1));
15         else
16             echo "$file wrong $lang $i/$num_lang - processing ";
17             xmlstarlet ed -L -N dc="http://purl.org/dc/elements/1.1/" -d "//dc:language[$i]" "
                $file";
18         fi;
19         num_lang='xmlstarlet sel -N dc="http://purl.org/dc/elements/1.1/" -T -t -v "//dc:
            language" -n "$file" | wc -l';
20
21     done;
22 done;

```

Listing 4: Filter-DeEn-oai.sh: Shell-Skript, das anderssprachige Knoten in den von E-LIS geharvesteten XML-Dateien entfernt

```

1  xslt = e-lis.xsl
2  institution = "E-LIS"
3  collection = "E-LIS"
4  track_changes = 1
5  solr_core = "biblio"

```

Listing 5: Angepasste e-lis.properties-Datei

```

1  <!-- Urspruengliche dspace.xsl -->
2  <field name="id">
3      <xsl:value-of select="//identifier"/>
4  </field>
5
6  <xsl:if test="//dc:subject">
7      <xsl:for-each select="//dc:subject">
8          <xsl:if test="string-length() > 0">
9              <field name="topic">
10                 <xsl:value-of select="normalize-space()"/>
11             </field>
12         </xsl:if>
13     </xsl:for-each>
14 </xsl:if>

```

```

15
16 <!-- Aenderung fuer e-lis.xsl -->
17 <field name="id">
18   <xsl:value-of select="concat('e-lis', substring-after(//identifier,'org:'))"/>
19 </field>
20
21 <!-- SUBJECT -->
22 <xsl:if test="//dc:subject">
23   <xsl:for-each select="//dc:subject">
24     <xsl:if test="string-length() > 0">
25       <field name="topic">
26         <xsl:value-of select="substring(substring-before(//dc:subject,':'),4)"/>
27       </field>
28     </xsl:if>
29   </xsl:for-each>
30 </xsl:if>

```

Listing 6: Angepasstes e-lis.xsl-Stylesheet

```

1 #Fuer erste Seite der Trefferliste
2 base_url="http://primo.kobv.de:80/primo_library/libweb/action/login.do?afterPDS=true&[
   redirect_und_searchscope]&vl%28173155009UI0%29=lsr19i&tab=default%5Ftab&ct=Previous+
   Page&mode=Basic&indx=11&vl%28421939160UI1%29=all%5Fitems&vl%28freeText0%29=AN+*&fn=
   search&pds_handle=GUEST";
3 wget "$base_url" -O "RVK.html";
4
5 grep 'reportBibTip' "RVK.html" | awk '{gsub(/.*reportBibTip/, ""); print}' | cut -d '"' -
   f 2 >> PNK.identifiers;
6 max_results=$(grep '<!-- (' RVK.html | awk '{gsub(/.*\(/, ""); print}' | cut -d ")" -f 1';
7 last_pos=$((max_results-10));
8 pos=1;
9 rm -rf "RVK.html";
10
11 #Fuer alle weiteren Seiten der Trefferliste
12 while [ $pos -lt $last_pos ]; do
13
14     scroll_url="http://primo.kobv.de:80/primo_library/libweb/action/login.do?afterPDS=
       true&[redirect_und_searchscope]&vl%28173155010UI0%29=lsr19&tab=default%5Ftab&
       dstmp=1354370753889&ct=Next+Page&mode=Basic&indx=$pos\&vl%28421939160UI1%29=all
       %5Fitems&fromLogin=true&vl%28freeText0%29=AN+*&fn=search&pds_handle=GUEST"
15
16     echo "Downloading $pos of $max_results";
17
18     wget "$scroll_url" -O "RVK.html" -o "Resultlists.log";
19     pos=$((pos+10));
20     grep 'reportBibTip' "RVK.html" | awk '{gsub(/.*reportBibTip/, ""); print}' | cut -d '"'
       -f 2 >> PNK.identifiers;
21
22     rm -rf "RVK.html";
23     rm -rf "Resultlists.log";
24
25 done

```

Listing 7: Harvest-Resultlists-AN-Primo.sh: Shell-Skript, das alle Identifier aus einer Trefferliste von Primus filtert (URLs in Variablen zur Übersichtlichkeit gekürzt)

```

1  #!/bin/bash
2
3  i=1;
4  max_id='wc -l PNX.identifiers';
5
6  while read line; do
7
8      id='echo "$line" | awk '{gsub(/.*aleph/, ""); print}';
9
10     short_url="http://info.ub.hu-berlin.de/primus/$id";
11     result_url='curl -L "$short_url" | grep 'location' | awk '{gsub(/.*goto\/\/, ""); print}'
        | cut -d '"' -f 1';
12     wget $result_url -O "AN-PNX/result-$id.html" -o "result.log";
13
14     temp_entry_url='grep 'href="display.do' "AN-PNX/result-$id.html" | awk '{gsub(/.*href=
        /, ""); print}' | cut -d '"' -f 1 | awk '{gsub("\\&", "\\&"); print}' | sort |
        uniq | grep -v 'gathStat';
15
16     entry_url="http://vs30.kobv.de/primus_library/libweb/action/$temp_entry_url&showPnx=true
        ";
17     wget $entry_url -O "AN-PNX/entry-$id.html" -o "result.log";
18
19     pnx_url='grep 'location' "AN-PNX/entry-$id.html" | awk '{gsub(/.*goto\/\/, ""); print}' |
        cut -d '"' -f 1';
20     echo -e "\e[00;31mDownloading $i of $max_id\e[00m";
21     wget $pnx_url -O "AN-PNX/PNX-$id.html" -o "result.log";
22
23     rm -rf "AN-PNX/result-$id.html";
24     rm -rf "AN-PNX/entry-$id.html";
25     rm -rf "result.log";
26
27     i=$((i+1));
28
29 done < PNX.identifiers

```

Listing 8: Harvest-PNX.sh: Shell-Skript, das alle Einträge in der Primus-Trefferliste im PNX-Format herunterlädt

```

1  #!/bin/bash
2
3  echo "Specify XPath"
4  read xpath;
5  # Read user input: xpath to language node
6
7  for file in *xml; do
8
9      #Count number of languages
10     num_lang='xmlstarlet sel -T -t -v "count($xpath)" -n $file';
11     echo $num_lang;
12
13     if [ $num_lang -eq 0 ]; then
14         mv "$file" "NotUseful/$file";
15     fi;
16

```

```

17 if [ $num_lang -eq 1 ]; then
18   lang='xmlstarlet sel -T -t -v "$xpath" -n "$file"';
19   if [ "$lang" == "ger" ] || [ "$lang" == "eng" ]; then
20     echo "$file right $lang - do nothing";
21   else
22     echo "$file wrong $lang - move";
23     mv "$file" "NotUseful/$file";
24   fi;
25 fi;
26
27 if [ $num_lang -gt 1 ]; then
28   i=1;
29   while [ $i -le $num_lang ]; do
30     num_lang='xmlstarlet sel -T -t -v "count($xpath)" "$file"';
31     lang='xmlstarlet sel -T -t -v "$xpath[$i]" -n "$file"';
32     if [ "$lang" == "ger" ] || [ "$lang" == "eng" ]; then
33       echo "$file right $lang $i/$num_lang - do nothing";
34       i=$((i+1));
35     else
36       echo "$file wrong $lang $i/$num_lang - processing - ";
37       xmlstarlet ed -L -d "$xpath[$i]" "$file";
38       fi;
39       num_lang='xmlstarlet sel -T -t -v "count($xpath)" -n "$file" ' ;
40       done;
41 fi;
42
43 done

```

Listing 9: Filter-XML-Languages.sh: Shell-Skript, das alle nicht-deutsch- oder englischsprachigen XML-Dateien aus einer Kollektion entfernt und ggf. anderssprachige Knoten aus mehrsprachigen XML-Dateien entfernt

```

1 <xsl:stylesheet version="1.0"
2   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:php="http://php.net/xsl"
6   xmlns:xlink="http://www.w3.org/2001/XMLSchema-instance">
7   <xsl:output method="xml" indent="yes" encoding="utf-8"/>
8   <xsl:param name="institution">My University</xsl:param>
9   <xsl:param name="collection">Primus</xsl:param>
10  <xsl:template match="/">
11    <add>
12      <doc>
13
14        <!-- ID -->
15        <field name="id">
16          <xsl:value-of select="//recordid"/>
17        </field>
18
19        <!-- RECORDTYPE -->
20        <field name="recordtype">primus</field>
21
22        <!-- ALLFIELDS -->

```

```

23 <field name="allfields">
24   <xsl:value-of select="normalize-space(string(//record))"/>
25 </field>
26
27 <!-- INSTITUTION -->
28 <field name="institution">
29   <xsl:value-of select="$institution" />
30 </field>
31
32 <!-- COLLECTION -->
33 <field name="collection">
34   <xsl:value-of select="$collection" />
35 </field>
36
37 <!-- LANGUAGE -->
38 <xsl:if test="//facets/language">
39   <xsl:for-each select="//facets/language">
40     <xsl:if test="string-length() > 0">
41       <field name="language">
42         <xsl:value-of select="php:function('VuFind::mapString', normalize-space(string
43           ()), 'language_map.properties')"/>
44       </field>
45     </xsl:if>
46   </xsl:for-each>
47 </xsl:if>
48
49 <!-- SUBJECT -->
50 <xsl:if test="//search/subject">
51   <xsl:for-each select="//search/subject">
52     <xsl:if test="string-length() > 0">
53       <field name="topic">
54         <xsl:value-of select="normalize-space()"/>
55       </field>
56     </xsl:if>
57   </xsl:for-each>
58 </xsl:if>
59
60 <!-- TYPE -->
61 <xsl:if test="//display/type">
62   <field name="format">
63     <xsl:value-of select="php:function('VuFind::mapString', normalize-space(string(//
64       display/type)), 'format_map.properties')"/>
65   </field>
66 </xsl:if>
67
68 <!-- AUTHOR -->
69 <xsl:if test="//display/contributor">
70   <xsl:for-each select="//display/contributor">
71     <xsl:if test="normalize-space()">
72       <xsl:if test="position()=1">
73         <field name="author">
74           <xsl:value-of select="normalize-space()"/>
75         </field>
76       <field name="author-letter">
77         <xsl:value-of select="normalize-space()"/>
78       </field>

```

```

77     </xsl:if>
78     <xsl:if test="position()>1">
79         <field name="author2">
80             <xsl:value-of select="normalize-space()" />
81         </field>
82     </xsl:if>
83 </xsl:if>
84 </xsl:for-each>
85 </xsl:if>
86
87     <!-- TITLE -->
88     <xsl:if test="//display/title[normalize-space()]">
89         <field name="title">
90             <xsl:value-of select="//display/title[normalize-space()]" />
91         </field>
92         <field name="title_short">
93             <xsl:value-of select="//search/title[1][normalize-space()]" />
94         </field>
95         <field name="title_full">
96             <xsl:value-of select="//display/title[normalize-space()]" />
97         </field>
98         <field name="title_sort">
99             <xsl:value-of select="php:function('VuFind::stripArticles', string(//dc:title[
100                 normalize-space()]))" />
101         </field>
102     </xsl:if>
103
104     <!-- PUBLISHER -->
105     <xsl:if test="//display/publisher[normalize-space()]">
106         <field name="publisher">
107             <xsl:value-of select="//display/publisher[normalize-space()]" />
108         </field>
109     </xsl:if>
110
111     <!-- PUBLISHDATE -->
112     <xsl:if test="//display/creationdate">
113         <field name="publishDate">
114             <xsl:value-of select="substring(//display/creationdate, 1, 4)" />
115         </field>
116         <field name="publishDateSort">
117             <xsl:value-of select="substring(//display/creationdate, 1, 4)" />
118         </field>
119     </xsl:if>
120
121     <!-- URL -->
122     <xsl:if test="//sourcerecordid">
123         <field name="url">
124             <xsl:value-of select="concat('http://info.ub.hu-berlin.de/primus/', //
125                 sourcerecordid)" />
126         </field>
127     </xsl:if>
128
129     <!-- ISBN -->
130     <xsl:if test="//search/isbn">
131         <field name="isbn">
132             <xsl:value-of select="//search/isbn[1]" />
133         </field>
134     </xsl:if>

```



```

131     </field>
132 </xsl:if>
133
134 </doc>
135 </add>
136 </xsl:template>
137 </xsl:stylesheet>

```

Listing 10: primus.xsl: XSLT-Stylesheet für PNX-Datensätze

```

1  #!/bin/bash
2
3  current_pos=0;
4  max_pos='curl "https://portal.dnb.de/opac.htm?method=showPreviousRecord&currentResultId=
   ddc%3D02*%26any&currentPosition=0" | grep 'class="amount"' | awk '{gsub("\t","");
   print}' | uniq | cut -d ">" -f 2 | cut -d "<" -f 1 | awk '{print $NF}'';
5
6  while [ $current_pos -lt $max_pos ];
7
8      echo "Downloading File $current_pos of $max_pos"; do
9
10     id='curl "https://portal.dnb.de/opac.htm?method=showPreviousRecord&currentResultId=ddc
       %3D02*%26any&currentPosition=$current_pos" | grep 'requestMarcXml' | awk '{gsub(/.*
       idn=/,""); gsub("\".*",""); print}'';
11     wget "https://portal.dnb.de/opac.htm?method=requestMarcXml&idn=$id" -O $id.xml;
12     current_pos=$((current_pos+1));
13
14 done

```

Listing 11: Harvest-DNB.sh: Shell-Skript, das MARC21-XML-Datensätze vom Katalog der DNB herunterlädt

```

1  ### Feld 008/35-37 wird indexiert, Institution original ###
2  institution = "MyInstitution"
3  building = "Library A"
4
5  language = 008[35-37]:041a:041d:041h:041j, language_map.properties
6  format = custom, getFormat, format_map.properties
7
8  ### Feld 008/35-37 wird nicht indexiert, Institution umbenannt, Building entfernt ###
9
10 institution = "Deutsche Nationalbibliothek"
11
12 language = 041a:041d:041j, language_map.properties
13 format = custom, getFormat, format_map.properties

```

Listing 12: marc.properties: MARC-Feld 008/35-37 soll nicht indexiert werden, DNB als Institution eintragen (Ausschnitt), Building entfernen

```

1  #!/bin/bash
2

```

```

3  # Dieses Skript harvestet alle Links zu einzelnen Datensätzen bei SpringerLink.com (
    link.springer.com) in der Disziplin "Database Management & Information Retrieval".
    Die Datensätze an sich werden dabei noch nicht heruntergeladen.
4
5  current_result=1;
6  current_page=1;
7
8  start_url="http://link.springer.com/search?facet-discipline=%22Computer+Science%22&facet
    -sub-discipline=%22Database+Management+%26+Information+Retrieval%22&query=&facet-
    language=%22En%22";
9  echo -e "\e[00;31mDownloading $current_result-$((current_result+9)) of $max_results\e[00
    m";
10
11 curl "$start_url" | grep 'href="/book\|href="/chapter\|href="/referenceworkentry\|href="
    /article\|href="/bookseries\|href="/referencework\|href="/journal' | grep -v "
    lookinside" | awk '{gsub(/.*href="\/, ""); print}' | awk '{gsub(/\\".*\/, ""); print}' |
    sort | uniq >> SpringerLink.en.identifiers;
12
13 wget "$start_url" -O temp.html;
14 max_results_line='grep -n 'class="number-of-search-results-and-search-terms"' temp.html
    | cut -d ":" -f 1';
15 max_results='sed -n "$((max_results_line+1))"p' temp.html | awk '{gsub("<strong>", "");
    gsub("</strong>", ""); gsub(",", ""); print}';
16 rm -rf temp.html;
17
18 current_result=$((current_result+10));
19 current_page=$((current_page+1));
20
21 while [ $current_result -le $max_results ]; do
22
23     scroll_url="http://link.springer.com/search/page/$current_page?facet-language=%22En%22&
        facet-sub-discipline=%22Database+Management+%26+Information+Retrieval%22&facet-
        discipline=%22Computer+Science%22&query="
24
25     echo -e "\e[00;31mDownloading $current_result-$((current_result+9)) of $max_results\e
        [00m";
26
27     curl "$scroll_url" | grep 'href="/book\|href="/chapter\|href="/referenceworkentry\|href
        ="/article\|href="/bookseries\|href="/referencework\|href="/journal' | grep -v "
        lookinside" | awk '{gsub(/.*href="\/, ""); print}' | awk '{gsub(/\\".*\/, ""); print}'
        | sort | uniq >> SpringerLink.en.identifiers;
28
29     current_result=$((current_result+10));
30     current_page=$((current_page+1));
31
32 done;
33
34 sort SpringerLink.en.identifiers | uniq > Sorted_SpringerLink.en.identifiers

```

Listing 13: Harvest-SpringerLink-Resultlists.en.sh: Shell-Skript, das alle Links zu Einträgen aus einer Resultatsliste von SpringerLink herunterlädt

```

1  #!/bin/bash
2
3  base_url="http://link.springer.com";

```

```

4 num_files='wc -l Articles.SpringerLink.en.identifiers';
5 at_file=1;
6
7 while read line; do
8
9     echo -e "\e[00;31mDownloading File $at_file of $num_files\e[00m";
10
11     filename='echo $line | awk '{gsub("/a","a"); gsub("/", "_"); print}';
12     wget "$base_url$line" -O "Articles/$filename.html" -a "Articles.SpringerLink.log";
13
14     record_type='echo "$line" | cut -d "/" -f 2';
15     title='grep 'id="abstract-about-title' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"
16         "); print}' | awk '{gsub(/<.*"/,""); print}';
17     subtitle='grep 'id="abstract-about-book-subtitle' "Articles/$filename.html" | awk '{
18         gsub(/.*\"/>,""); print}' | awk '{gsub(/<.*"/,""); print}';
19     journal='grep 'id="abstract-about-publication' "Articles/$filename.html" | awk '{gsub
20         (/.*\"/>,""); print}' | awk '{gsub(/<.*"/,""); print}';
21     issue='grep 'id="abstract-about-issue' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"
22         "); print}' | awk '{gsub(/<.*"/,""); print}' | cut -d "," -f 2 | awk '{gsub(" Issue
23         ",""); gsub(" ",""); print}';
24     volume='grep 'id="abstract-about-issue' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"
25         "); print}' | awk '{gsub(/<.*"/,""); print}' | cut -d "," -f 1 | awk '{gsub("Volume
26         ",""); gsub(" ",""); print}';
27     pages='grep 'id="abstract-about-issue' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"
28         "); print}' | awk '{gsub(/<.*"/,""); print}' | cut -d "," -f 3 | awk '{gsub(" pp ","
29         "); print}';
30     DOI='grep 'id="abstract-about-doi' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"");
31         print}' | awk '{gsub(/<.*"/,""); print}';
32     ISSN_print='grep 'id="abstract-about-issn' "Articles/$filename.html" | awk '{gsub(/.*\
33         ">",""); print}' | awk '{gsub(/<.*"/,""); print}';
34     ISSN_online='grep 'id="abstract-about-electronic-issn' "Articles/$filename.html" | awk
35         '{gsub(/.*\"/>,""); print}' | awk '{gsub(/<.*"/,""); print}';
36     publisher='grep 'id="abstract-about-publisher' "Articles/$filename.html" | awk '{gsub
37         (/.*\"/>,""); print}' | awk '{gsub(/<.*"/,""); print}';
38     cover_date='grep 'id="abstract-about-cover-date' "Articles/$filename.html" | awk '{gsub
39         (/.*\"/>,""); print}' | awk '{gsub(/<.*"/,""); print}';
40
41     ### START: Get Names of Authors ###
42
43     author_lines='grep -n 'li itemprop="author' "Articles/$filename.html" | cut -d ":" -f
44         1';
45     declare -a author;
46
47     for linia in $author_lines; do
48
49         author_temp='sed -n "${(linia+1)}p' "Articles/$filename.html" | awk '{gsub(/.*\"/>,"
50             "); print}' | awk '{gsub(/<.*"/,""); print}';
51         author=("${author[@]}" "$author_temp");
52
53     done;
54
55     ### END: Get Names of Authors ###
56
57     ### START: Get Topics/Subjects ###
58
59     subject_lines='grep -n 'href="/search?facet=subject' "Articles/$filename.html" | cut -d

```

```

    ":" -f 1';
44 declare -a subject;
45
46 for linia in $subject_lines; do
47
48     subject_temp='sed -n "$linia"p' "Articles/$filename.html" | awk '{gsub(/.*\>/,"");
49         print}' | awk '{gsub(/<.*\/,""); print}';
50     subject=("${subject[@]} " $subject_temp");
51
52 done;
53
54 ### END: Get Topics/Subjects ###
55
56     ### START: Write XML ###
57
58 echo -e "\e[00;31mCreating $filename.xml\e[00m";
59
60 echo -e "<record>" >> "Articles/$filename.xml";
61 echo -e "\t <type>$record_type</type>" >> "Articles/$filename.xml";
62 echo -e "\t <title>$title</title>" >> "Articles/$filename.xml";
63 echo -e "\t <subtitle>$subtitle</subtitle>" >> "Articles/$filename.xml";
64 echo -e "\t <journal>$journal</journal>" >> "Articles/$filename.xml";
65 echo -e "\t <volume>$volume</volume>" >> "Articles/$filename.xml";
66 echo -e "\t <issue>$issue</issue>" >> "Articles/$filename.xml";
67 echo -e "\t <pages>$pages</pages>" >> "Articles/$filename.xml";
68 echo -e "\t <doi>$DOI</doi>" >> "Articles/$filename.xml";
69 echo -e "\t <issn_print>$ISSN_print</issn_print>" >> "Articles/$filename.xml";
70 echo -e "\t <issn_online>$ISSN_online</issn_online>" >> "Articles/$filename.xml";
71 echo -e "\t <url>$base_url$line</url>" >> "Articles/$filename.xml";
72 echo -e "\t <publisher>$publisher</publisher>" >> "Articles/$filename.xml";
73 echo -e "\t <coverdate>$cover_date</coverdate>" >> "Articles/$filename.xml";
74 echo -e "\t <language>en</language>" >> "Articles/$filename.xml"
75
76 echo -e "\t <creators>" >> "Articles/$filename.xml";
77 i=0;
78 num_authors='echo "${#author[*]}"';
79 while [ $i -le $(num_authors-1) ]; do
80     echo -e "\t\t <creator>${author[$i]}</creator>" >> "Articles/$filename.xml";
81     i=$((i+1));
82 done;
83 echo -e "\t </creators>" >> "Articles/$filename.xml";
84
85 i=0;
86 num_subjects='echo "${#subject[*]}"';
87 echo -e "\t <subjects>" >> "Articles/$filename.xml";
88 while [ $i -le $(num_subjects-1) ]; do
89     echo -e "\t\t <subject>${subject[$i]}</subject>" >> "Articles/$filename.xml";
90     i=$((i+1));
91 done;
92 echo -e "\t </subjects>" >> "Articles/$filename.xml";
93
94 echo -e "</record>" >> "Articles/$filename.xml";
95
96 echo "Finishing XML";
97 unset editor[*];

```

```

98 unset subject[*];
99 unset author[*];
100
101 rm -rf "Articles/$filename.html";
102 at_file=$((at_file+1));
103
104 done < Articles.SpringerLink.en.identifiers

```

Listing 14: Harvest-SpringerLink-Articles.sh: Shell-Skript, das HTML-Quelltexte von SpringerLink analysiert, Metadaten extrahiert und diese in einer XML-Datei speichert

```

1 <xsl:stylesheet version="1.0"
2   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:php="http://php.net/xsl"
6   xmlns:xlink="http://www.w3.org/2001/XMLSchema-instance">
7   <xsl:output method="xml" indent="yes" encoding="utf-8"/>
8   <xsl:param name="institution">My University</xsl:param>
9   <xsl:param name="collection">SpringerLink</xsl:param>
10  <xsl:template match="/">
11    <add>
12      <doc boost="1.5">
13
14        <!-- ID -->
15        <field name="id">
16          <xsl:value-of select="//id"/>
17        </field>
18
19        <!-- RECORDTYPE -->
20        <field name="recordtype">SpringerLink-Article</field>
21
22        <!-- ALLFIELDS -->
23        <field name="allfields">
24          <xsl:value-of select="normalize-space(string(//*))"/>
25        </field>
26
27        <!-- INSTITUTION -->
28        <field name="institution">
29          <xsl:value-of select="$institution" />
30        </field>
31
32        <!-- COLLECTION -->
33        <field name="collection">
34          <xsl:value-of select="$collection" />
35        </field>
36
37        <!-- LANGUAGE -->
38        <xsl:if test="//language">
39          <xsl:for-each select="//language">
40            <xsl:if test="string-length() > 0">
41              <field name="language">
42                <xsl:value-of select="php:function('VuFind::mapString', normalize-space(string(
43                  ()), 'language_map.properties')"/>

```

```

44     </xsl:if>
45 </xsl:for-each>
46 </xsl:if>
47
48 <!-- SUBJECT -->
49 <xsl:if test="//subject">
50   <xsl:for-each select="//subject">
51     <xsl:if test="string-length() > 0">
52       <field name="topic">
53         <xsl:value-of select="normalize-space()"/>
54       </field>
55     </xsl:if>
56   </xsl:for-each>
57 </xsl:if>
58
59 <!-- TYPE -->
60 <xsl:if test="//type">
61   <field name="format">
62     <xsl:value-of select="php:function('VuFind::mapString', string(//type), '
        format_map.properties')" />
63   </field>
64 </xsl:if>
65
66 <!-- AUTHOR -->
67 <xsl:if test="//creator">
68   <xsl:for-each select="//creator">
69     <xsl:if test="normalize-space()">
70       <xsl:if test="position()=1">
71         <field name="author">
72           <xsl:value-of select="normalize-space()"/>
73         </field>
74         <field name="author-letter">
75           <xsl:value-of select="normalize-space()"/>
76         </field>
77       </xsl:if>
78       <xsl:if test="position()>1">
79         <field name="author2">
80           <xsl:value-of select="normalize-space()"/>
81         </field>
82       </xsl:if>
83     </xsl:if>
84   </xsl:for-each>
85 </xsl:if>
86
87 <!-- TITLE -->
88 <xsl:if test="//title[normalize-space()]">
89   <field name="title">
90     <xsl:value-of select="//title[normalize-space()"/>
91   </field>
92   <field name="title_full">
93     <xsl:value-of select="//title[normalize-space()"/>
94   </field>
95   <field name="title_sort">
96     <xsl:value-of select="php:function('VuFind::stripArticles', string(//dc:title[
        normalize-space()]))"/>
97   </field>

```

```

98     </xsl:if>
99
100    <!-- Journal-Angaben -->
101    <xsl:if test="//journal[normalize-space()]">
102        <field name="series">
103            <xsl:value-of select="concat(//journal , ' ' , //volume , ' ( ' , substring(//
              coverdate,1,4) , '), Nr. ' , //issue , ' , ' , //pages)"/>
104        </field>
105    </xsl:if>
106
107    <!-- Publisher -->
108    <xsl:if test="//publisher">
109        <field name="publisher">
110            <xsl:value-of select="//publisher"/>
111        </field>
112    </xsl:if>
113
114    <!-- PUBLISHDATE -->
115    <xsl:if test="//coverdate">
116        <field name="publishDate">
117            <xsl:value-of select="substring(//coverdate , 1, 4)"/>
118        </field>
119        <field name="publishDateSort">
120            <xsl:value-of select="substring(//coverdate , 1, 4)"/>
121        </field>
122    </xsl:if>
123
124    <!-- URL -->
125    <xsl:if test="//doi">
126        <field name="url">
127            <xsl:value-of select="concat('http://dx.doi.org/',//doi)" />
128        </field>
129    </xsl:if>
130
131    <!-- ISSN -->
132    <xsl:if test="//issn\_online">
133        <field name="issn">
134            <xsl:value-of select="//issn\_online"/>
135        </field>
136    </xsl:if>
137
138    </doc>
139    </add>
140 </xsl:template>
141 </xsl:stylesheet>

```

Listing 15: springerlink.xsl: XSLT-Stylesheet für die Zeitschriftenartikel von SpringerLink

1	Book	=	Buch
2	Newspaper	=	Newspaper
3	Journal	=	Zeitschrift
4	Article	=	Zeitschriftenartikel
5	ConferencePaper	=	Konferenzbeitrag
6	Presentation	=	Praesentation

```

7 Preprint          = Preprint
8 Thesis            = Dissertation
9 Report            = Report
10 Review            = Review
11 journal           = Zeitschrift
12 Guide/Manual      = Buch
13 Bibliography      = Bibliographie
14 microform         = Mikroform
15 E-Book            = eBook
16 BookChapter       = Buchkapitel
17 JournalArticle    = Zeitschriftenartikel
18 Project/BusinessPlan = Projektbericht
19 ConferenceProceeding = Konferenzband
20 ConferencePoster   = Konferenzposter
21 Newspaper/MagazineArticle = Zeitungsartikel

```

Listing 16: `format_map.properties`: Angepasste Translation Map für Formate

```

1 <!-- Standard -->
2 <filter class="solr.SnowballPorterFilterFactory" language="English"/>
3
4 <!-- LISard -->
5 <filter class="solr.SnowballPorterFilterFactory" language="German2"/>
6
7 <!-- Diese Subnode muss in den Nodes <analyzer type="index"> und <analyzer type="query">
   geaendert werden -->

```

Listing 17: `schema.xml`: Auswahl des Stemmers

```

1 ; Standard
2 [Results]
3 institution      = Institution
4 building          = Library
5 format           = Format
6 callnumber-first = "Call Number"
7 authorStr        = Author
8 language         = Language
9 genre_facet      = Genre
10 era_facet        = Era
11 geographic_facet = Region
12 publishDate      = "adv_search_year";
13
14 ; Fuer LISard angepasst
15 [Results]
16 institution      = Ressource ; equals Institution
17 format           = Format
18 authorStr        = Author
19 language         = Language
20 topic            = Subject
21 publishDate      = "adv_search_year"

```

Listing 18: `facets.ini`: Gegenüberstellung der voreingestellten und für LISard angepassten `facets.ini`



```

1 <!-- result.tpl: Aufruf von Signatur und Standort -->
2 <!--
3 {if $summAjaxStatus}
4 <b>{translate text='Call Number'}:</b> <span id="callnumber{$summId|escape}">{translate
   text='Loading'}</span><br>
5 <b>{translate text='Located'}:</b> <span id="location{$summId|escape}">{translate text
   ='Loading'}</span>
6 {elseif !empty($summCallNo)}
7 <b>{translate text='Call Number'}:</b> {$summCallNo|escape}
8 {/if}
9 -->
10
11 <!-- result.tpl: Aufruf der Verfgbaruekeit -->
12 <!--
13 <div class="status" id="status{$summId|escape}">
14   <span class="unknown" style="font-size: 8pt;">{translate text='Loading'}...</span>
15 </div>
16 -->
17
18 <!-- holdings.tpl: Aufruf von Signatur, Standort und Verfuegbarkeit -->
19 <!--
20 {foreach from=$holdings item=holding key=location}
21 <h3>{$location|translate|escape}</h3>
22 <table cellpadding="2" cellspacing="0" border="0" class="citation" summary="{translate
   text='Holdings details from'} {translate text=$location}">
23   {if $holding.0.callnumber}
24   <tr>
25     <th>{translate text="Call Number"}:</th>
26     <td>{$holding.0.callnumber|escape}</td>
27   </tr>
28   {/if}
29   {if $holding.0.summary}
30   <tr>
31     <th>{translate text="Volume Holdings"}:</th>
32     <td>
33       {foreach from=$holding.0.summary item=summary}
34       {$summary|escape}<br>
35       {/foreach}
36     </td>
37   </tr>
38   {/if}
39   {if $holding.0.notes}
40   <tr>
41     <th>{translate text="Notes"}:</th>
42     <td>
43       {foreach from=$holding.0.notes item=data}
44       {$data|escape}<br>
45       {/foreach}
46     </td>
47   </tr>
48   {/if}
49   {foreach from=$holding item=row}
50     {if $row.barcode != ""}
51     <tr>
52       <th>{translate text="Copy"} {$row.number|escape}</th>
53       <td>

```

```

54     {if $row.reserve == "Y"}
55     {translate text="On Reserve - Ask at Circulation Desk"}
56     {else}
57     {if $row.availability}
58     {* Begin Available Items (Holds) *}
59     <div>
60     <span class="available">{translate text="Available"}</span>
61     {if $row.link}
62     <a class="holdPlace{if $row.check} checkRequest{/if}" href="{ $row.link|escape}"><
        span>{if !$row.check}{translate text="Place a Hold"}{else}{translate text="
        Check Hold"}{/if}</span></a>
63     {/if}
64     </div>
65     {else}
66     {* Begin Unavailable Items (Recalls) *}
67     <div>
68     <span class="checkedout">{translate text=$row.status}</span>
69     {if $row.returnDate} <span class="statusExtra">{$row.returnDate|escape}</span>{/if
        }
70     {if $row.duedate}
71     <span class="statusExtra">{translate text="Due"}: {$row.duedate|escape}</span>
72     {/if}
73     {if $row.requests_placed > 0}
74     <span>{translate text="Requests"}: {$row.requests_placed|escape}</span>
75     {/if}
76     {if $row.link}
77     <a class="holdPlace{if $row.check} checkRequest{/if}" href="{ $row.link|escape}"><
        span>{if !$row.check} translate text="Recall This"{else}{translate text="
        Check Recall"}{/if}</span></a>
78     {/if}
79     </div>
80     {/if}
81     {/if}
82 </td>
83 </tr>
84 {/if}
85 {/foreach}
86 </table>
87 {/foreach}
88 -->

```

Listing 19: Auskommentierte Stellen in den Dateien result.tpl und holdings.tpl

```

1  {* Your footer *}
2  <div><p><strong>{translate text='Search Options'}</strong></p>
3    <ul>
4      <li><a href="{ $path}/Search/History">{translate text='Search History'}</a></li>
5      <li><a href="{ $path}/Search/Advanced">{translate text='Advanced Search'}</a></li>
6    </ul>
7  </div>
8  <!-- Recherche Funktionen:
9  <div><p><strong>{translate text='Find More'}</strong></p>
10    <ul>
11      <li><a href="{ $path}/Browse/Home">{translate text='Browse the Catalog'}</a></li>
12      <li><a href="{ $path}/AlphaBrowse/Home">{translate text='Browse Alphabetically'}</a><
        /li>

```

```

13     <li><a href="{ $path }/Search/Reserves">{translate text='Course Reserves'}</a></li>
14     <li><a href="{ $path }/Search/NewItem">{translate text='New Items'}</a></li>
15 </ul>
16 </div>
17 -->
18 <div><p><strong>{translate text='Need Help?'}</strong></p>
19     <ul>
20         <li><a href="{ $url }/Help/Home?topic=search" onClick="window.open('{ $url }/Help/Home?
           topic=search', 'Help', 'width=625, height=510'); return false;">{translate text=
           'Search Tips'}</a></li>
21 <!-- Hilfe-Funtionen:
22     <li><a href="#">{translate text='Ask a Librarian'}</a></li>
23     <li><a href="#">{translate text='FAQs'}</a></li>
24 -->
25     </ul>
26 </div>
27 <br clear="all">

```

Listing 20: footer.tpl: Bestimmte Recherche- und Hilfefunktionen deaktivieren

```

1 [Content]
2 coverimages      = Google,OpenLibrary
3 previews         = Google,OpenLibrary,HathiTrust

```

Listing 21: config.ini: Buch-Cover und weitere Informationen laden

```

1 [Autocomplete]
2 enabled = true
3 default_handler = SolrAutocomplete
4
5 [Autocomplete_Types]
6 Title = "SolrAutocomplete:Title"
7 JournalTitle = "SolrAutocomplete:JournalTitle"
8 Subject = "SolrAutocomplete:Subject:topic"
9 tag = "TagAutocomplete"

```

Listing 22: searches.ini: Autovervollständigung für Suchanfragen einschalten

## **E DVD**

Die beiliegende DVD enthält:

- eine elektronische Ausfertigung dieser Arbeit im PDF-Format
- eine Kopie des VuFind-Verzeichnisses samt erstellter Skripte, geharvesteter Datensätze sowie dem kompletten LISard-Index

# Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorgelegte Arbeit eigenständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Stellen der Arbeit, die im Wortlaut oder wesentlichen Inhalt aus anderen Werken oder dem Internet entnommen wurden, mit genauen Quellenangaben kenntlich gemacht habe. Verwendete Informationen aus dem Internet sind den Gutachtern ggf. auf Nachfrage vollständig zur Verfügung zu stellen. Die Arbeit hat in dieser oder ähnlicher Form noch nicht im Rahmen einer anderen Prüfung vorgelegen.

Berlin, 23.12.2012

Ort, Datum

---

Oliver Pohl